Cloud Data Migration

User Guide

Issue 1

Date 2023-06-21

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property 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 Cloud 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.

Contents

1 Permissions Management

1.1 Creating a User and Granting CDM Permissions

This chapter describes how to use **Identity and Access Management (IAM)** to implement fine-grained permissions control for your CDM resources. With IAM, you can:

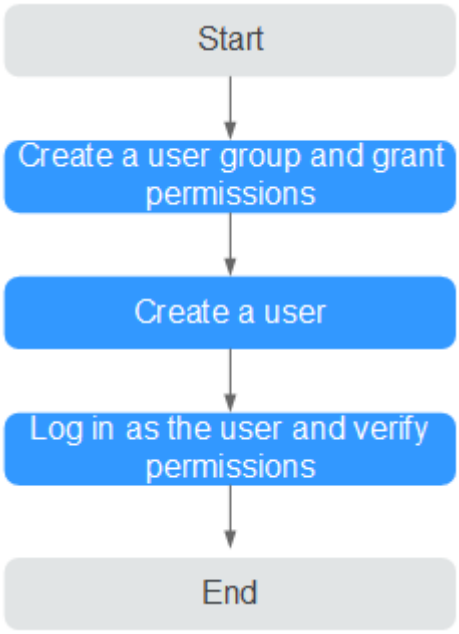
- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing CDM resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust a Huawei Cloud account or cloud service to perform efficient O&M on your CDM resources.

If your Huawei Cloud account does not require individual IAM users, skip this chapter.

This section describes the procedure for granting permissions (see **Figure 1-1**).

Process Flow

Figure 1-1 Process of granting CDM permissions



1. **Create a user group and assign permissions**

   Create a user group on the IAM console, and attach the **CDM ReadOnlyAccess** policy to the group.

2. **Create an IAM user.**

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Log in** and verify permissions.

   Log in to the CDM console by using the user created, and verify that the user only has read permissions for CDM.

   - Choose **Service List** > **Cloud Data Migration**. On the CDM console, view clusters. If no message appears indicating insufficient permissions to perform the operation, the **CDM ReadOnlyAccess** policy has already taken effect.
   - Choose any other service in **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **CDM ReadOnlyAccess** policy has already taken effect.

1.2 Creating a Custom Policy

Custom policies can be created to supplement the system-defined policies of CDM. For the actions that can be added to custom policies, see **Permissions Policies and Supported Actions**.

You can create custom policies in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.

- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. The following section contains examples of common CDM custom policies.

Example Custom Policies

- Example 1: Allowing users to create a CDM cluster

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "cdm:cluster:create"
         ]
      }
   ]
}
```

- Example 2: Denying CDM cluster deletion

  A policy with only "Deny" permissions must be used in conjunction with other policies to take effect. If the permissions assigned to a user contain both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

  The following method can be used if you need to assign permissions of the **CDM FullAccess** policy to a user but you want to prevent the user from deleting CDM clusters. Create a custom policy for denying CDM cluster deletion, and attach both policies to the group to which the user belongs. Then, the user can perform all operations on CDM resources except deleting CDM clusters. The following is an example of a deny policy:

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Deny",
         "Action": [
            "cdm:cluster:delete"
         ]
      }
   ]
}
```

- Example 3: Defining permissions for multiple services in a policy

  A custom policy can contain actions of multiple services that are of the global or project-level type. The following is an example policy containing actions of multiple services:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "cdm:cluster:list",
                "cdm:cluster:get",
                "ecs:*:get*",
                "ecs:*:list*",
                "vpc:*:get*",
                "vpc:*:list*",
                "evs:*:get*",
                "evs:*:list*",
                "bss:*:view*"
            ],
            "Effect": "Allow"
```



```
        }
    ]
}
```


2 Supported Data Sources

2.1 Supported Data Sources (2.9.3.300)

CDM provides the following migration modes which support different data sources:

- **Table/File migration** in the import of data into a data lake or migration of data to the cloud. For details, see **Data Sources Supported by Table/File Migration**.
- **Entire DB migration** in the import of data into a data lake or migration of data to the cloud. For details, see **Supported Data Sources in Entire DB Migration**.

NOTE

This section describes the data sources supported by CDM clusters of version 2.9.3.300. The supported data sources vary depending on the CDM cluster version.

Data Sources Supported by Table/File Migration

Table/File migration can migrate data in tables or files.

Table 2-1 describes the supported data sources.

Table 2-1 Supported data sources during table/file migration

Category	Source	Destination	Description
Data warehouse	Data Warehouse Service	• Data warehouse: GaussDB(DWS), Data Lake Insight (DLI), and MRS ClickHouse • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object-based storage: Object Storage Service (OBS) • Relational database: RDS for MySQL, RDS for PostgreSQL, RDS for SQL Server, MySQL, PostgreSQL, Microsoft SQL Server, and Oracle • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	The DWS physical machine management mode is not supported.
	Data Lake Insight (DLI)	• Data warehouse: GaussDB(DWS), Data Lake Insight (DLI), and MRS ClickHouse • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object-based storage: Object Storage Service (OBS) • Relational database: RDS for MySQL, RDS for PostgreSQL, RDS for SQL Server, MySQL, PostgreSQL, Microsoft SQL Server, and Oracle • NoSQL: CloudTable and MongoDB • Search: Elasticsearch and Cloud Search Service (CSS)	Recommended MongoDB version: 4.2
	MRS ClickHouse	Data warehouse: MRS ClickHouse and Data Lake Insight (DLI)	Recommended MRS ClickHouse version: 21.3.4.*X*

Category	Source	Destination	Description
Hadoop	MRS HDFS MRS HBase	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object-based storage: Object Storage Service (OBS) • Relational database: RDS for MySQL, RDS for PostgreSQL, RDS for SQL Server, MySQL, PostgreSQL, Microsoft SQL Server, and Oracle • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	• Supported by local storage. Only MRS Hive and MRS Hudi are supported in storage-compute decoupling scenarios. • Only MRS Hive is supported in Ranger scenarios. • Not supported if SSL is enabled for ZooKeeper • Recommended MRS HDFS versions: – 2.8.X – 3.1.X • Recommended MRS HBase versions: – 2.1.X – 1.3.X • MRS Hive and MRS Hudi 2.*x* versions are not supported. The following versions are recommended: – 1.2.X – 3.1.X
	MRS Hive	• Data warehouse: GaussDB(DWS), Data Lake Insight (DLI), and MRS ClickHouse • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object-based storage: Object Storage Service (OBS) • Relational database: RDS for MySQL, RDS for PostgreSQL, RDS for SQL Server, MySQL, PostgreSQL, Microsoft SQL Server, and Oracle • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	
	MRS Hudi	• Data warehouse: GaussDB(DWS) • Hadoop: MRS HBase	

Category	Source	Destination	Description
	FusionInsight HDFS FusionInsight HBase FusionInsight Hive	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object-based storage: Object Storage Service (OBS) • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	• FusionInsight cannot serve as the destination. • Supported only by local storage and not in storage-compute decoupling scenarios • Not supported by Ranger • Not supported if SSL is enabled for ZooKeeper • Recommended FusionInsight HDFS versions: – 2.8.X – 3.1.X • Recommended FusionInsight HBase versions: – 2.1.X – 1.3.X • Recommended FusionInsight Hive versions: – 1.2.X – 3.1.X

Category	Source	Destination	Description
	Apache HBase Apache Hive Apache HDFS	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object-based storage: Object Storage Service (OBS) • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	• Apache cannot serve as the destination. • Supported only by local storage and not in storage-compute decoupling scenarios • Not supported by Ranger • Not supported if SSL is enabled for ZooKeeper • Recommended Apache HBase versions: – 2.1.X – 1.3.X • Apache Hive 2.*x* versions are not supported. The following versions are recommended: – 1.2.X – 3.1.X • Recommended Apache HDFS versions: – 2.8.X – 3.1.X
Object storage	Object Storage Service (OBS)	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	• Object Storage Migration Service (OMS) is recommended for migration between object storage services. • Binary files cannot be imported to a database or NoSQL.

Category	Source	Destination	Description
File system	FTP SFTP	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS) • Object-based storage: Object Storage Service (OBS)	• The file system cannot serve as the destination. • Only text files such as CSV files can be migrated from FTP or SFTP servers to search services. Binary files cannot. • Only binary files can be migrated from FTP or SFTP servers to OBS. • obsutil is recommended for migrating data from HTTP servers to OBS. For details, see **Introduction to obsutil**.
	HTTP	Hadoop: MRS HDFS	

Category	Source	Destination	Description
Relational database	RDS for MySQL	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, MRS Hive, and MRS Hudi • Object-based storage: Object Storage Service (OBS) • NoSQL: CloudTable • Relational database: RDS for MySQL, RDS for PostgreSQL, and RDS for SQL Server • Search: Elasticsearch and Cloud Search Service (CSS)	• You are advised to use Data Replication Service (DRS) to migrate data between OLTP databases. • RDS for MySQL does not support the SSL mode. • Recommended Microsoft SQL Server version: 2005 or later • The KingBase database and GaussDB can be connected using the PostgreSQL connector. The supported source and destination are the same as those of the PostgreSQL data source.
	RDS for SQL Server RDS for PostgreSQL	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object-based storage: Object Storage Service (OBS) • NoSQL: CloudTable • Relational database: RDS for MySQL, RDS for PostgreSQL, and RDS for SQL Server • Search: Elasticsearch and Cloud Search Service (CSS)	
	MySQL PostgreSQL Oracle	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, MRS Hive, and MRS Hudi • Object-based storage: Object Storage Service (OBS) • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	
	Microsoft SQL Server	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object-based storage: Object Storage Service (OBS) • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	

Category	Source	Destination	Description
	SAP HANA	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS Hive	SAP HANA data sources have the following restrictions: • SAP HANA cannot serve as the destination. • Only the 2.00.050.00.1592305219 version is supported. • Only the Generic Edition is supported. • BW/4 FOR HANA is not supported. • Only database names, table names, and column names consisting of English letters are supported. Special characters such as spaces and symbols are not allowed. • The following data types are supported: date, digit, Boolean, and character (except SHORTTEXT). Other data types such as binary are not supported. • During migration, tables cannot be automatically created at the destination.

Category	Source	Destination	Description
	Database Sharding	• Data warehouse: Data Lake Insight (DLI) • Hadoop: MRS HBase and MRS Hive • Search: Elasticsearch and Cloud Search Service (CSS) • Object-based storage: Object Storage Service (OBS)	Database shards cannot serve as the destination.
	ShenTong	• Hadoop: MRS Hive and MRS Hudi	-
NoSQL	Distributed Cache Service (DCS)	Hadoop: MRS HDFS, MRS HBase, and MRS Hive	NoSQL except CloudTable cannot serve as the destination. For how to migrate data from Redis to DCS, see **Migrating Data from Self-Hosted Redis to DCS**.
	Redis		
	Document Database Service		
	MongoDB		
	CloudTable HBase	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object-based storage: Object Storage Service (OBS) • Relational database: RDS for MySQL, RDS for PostgreSQL, RDS for SQL Server, MySQL, PostgreSQL, Microsoft SQL Server, and Oracle • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	

Category	Source	Destination	Description
	Cassandra	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object-based storage: Object Storage Service (OBS) • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	
Message system	Data Ingestion Service (DIS)	Search: Cloud Search Service (CSS)	The message system cannot serve as the destination.
	Apache Kafka		
	DMS Kafka		
	MRS Kafka	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object-based storage: Object Storage Service (OBS) • Relational database: RDS for MySQL, RDS for PostgreSQL, and RDS for SQL Server • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	• MRS Kafka cannot serve as the destination. • Supported only by local storage and not in storage-compute decoupling scenarios • Not supported by Ranger • Not supported if SSL is enabled for ZooKeeper
Search	Elasticsearch	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object-based storage: Object Storage Service (OBS) • Relational database: RDS for MySQL, RDS for PostgreSQL, and RDS for SQL Server • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	Only the non-security mode is supported.
	Cloud Search Service (CSS)		You are advised to use Logstash to import data to CSS. For details, see **Using Logstash to Import Data to Elasticsearch**

NOTE

In the preceding table, the non-cloud data sources, such as MySQL, include on-premises MySQL, MySQL built on ECSs, or MySQL on the third-party cloud.

Supported Data Sources in Entire DB Migration

Entire DB migration is used when an on-premises data center or a database created on an ECS needs to be synchronized to a database service or big data service on the cloud. It is suitable for offline database migration but not online real-time migration.

Table 2-2 lists the data sources supporting entire DB migration using CDM.

Table 2-2 Supported data sources in entire DB migration

Category	Data Source	Read	Write	Description
Data warehouse	Data Warehouse Service	Supported	Supported	-
Hadoop (available only for local storage, and not for storage-compute decoupling, Ranger, or ZooKeeper for which SSL is enabled)	MRS HBase	Supported	Supported	Entire DB migration only to MRS HBase Recommended versions: • 2.1.X • 1.3.X
	MRS Hive	Supported	Supported	Entire DB migration only to a relational database 2.*x* versions are not supported. The following versions are recommended: • 1.2.X • 3.1.X
	FusionInsight HBase	Supported	Not supported	Recommended versions: • 2.1.X • 1.3.X

Category	Data Source	Read	Write	Description
	FusionInsight Hive	Supported	Not supported	Entire DB migration only to a relational database 2.*x* versions are not supported. The following versions are recommended: • 1.2.X • 3.1.X
	Apache HBase	Supported	Not supported	Recommended versions: • 2.1.X • 1.3.X
	Apache Hive	Supported	Not supported	Entire DB migration only to a relational database 2.*x* versions are not supported. The following versions are recommended: • 1.2.X • 3.1.X
	MRS Hudi	Supported	Supported	Supported only by local storage and in storage-compute decoupling scenarios 2.*x* versions are not supported. The following versions are recommended: • 1.2.X • 3.1.X

Category	Data Source	Read	Write	Description
Relational database	RDS for MySQL	Supported	Supported	Migration from OLTP to OLTP is not supported. In this scenario, you are advised to use the Data Replication Service (DRS).
	RDS for PostgreSQL	Supported	Supported	
	RDS for SQL Server	Supported	Supported	
	MySQL	Supported	Not supported	
	PostgreSQL	Supported	Not supported	
	Microsoft SQL Server	Supported	Not supported	
	Oracle	Supported	Not supported	

Category	Data Source	Read	Write	Description
	SAP HANA	Supported	Not supported	• Only the 2.00.050.00.1592305219 version is supported. • Only the Generic Edition is supported. • BW/4 FOR HANA is not supported. • Only database names, table names, and column names consisting of English letters are supported. Special characters such as spaces and symbols are not allowed. • The following data types are supported: date, digit, Boolean, and character (except SHORTTEXT). Other data types such as binary are not supported. • During migration, tables cannot be automatically created at the destination.
	Dameng database	Supported	Not supported	Only to DWS and Hive

Category	Data Source	Read	Write	Description
NoSQL	Distributed Cache Service (DCS)	Not supported	Supported	Only migration from MRS to DCS is supported.
	Document Database Service (DDS)	Supported	Supported	Only migration between DDS and MRS is supported.
	CloudTable	Supported	Supported	-

2.2 Supported Data Sources (2.9.2.200)

CDM provides the following migration modes which support different data sources:

- **Table/File migration** in the import of data into a data lake or migration of data to the cloud. For details, see **Data Sources Supported by Table/File Migration**.
- **Entire DB migration** in the import of data into a data lake or migration of data to the cloud. For details, see **Supported Data Sources in Entire DB Migration**.

NOTE

This section describes the data sources supported by CDM clusters of version 2.9.2.200. The supported data sources vary depending on the CDM cluster version.

Data Sources Supported by Table/File Migration

Table/File migration can migrate data in tables or files.

Table 2-3 describes the supported data sources.

Table 2-3 Supported data sources during table/file migration

Category	Source	Destination	Description
Data warehouse	GaussDB(DWS)	• Data warehouse: GaussDB(DWS), Data Lake Insight (DLI), and MRS ClickHouse • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object storage: Object Storage Service (OBS) • Relational database: RDS for MySQL, RDS for PostgreSQL, RDS for SQL Server, MySQL, PostgreSQL, Microsoft SQL Server, and Oracle • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	The DWS physical machine management mode is not supported.
	Data Lake Insight (DLI)		-
	MRS ClickHouse	Data warehouse: MRS ClickHouse and Data Lake Insight (DLI)	Recommended MRS ClickHouse version: 21.3.4.*X*

<table>
<tr><th>Category</th><th>Source</th><th>Destination</th><th>Description</th></tr>
<tr><td rowspan="3">Hadoop</td><td>MRS HDFS
MRS HBase</td><td>• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI)
• Hadoop: MRS HDFS, MRS HBase, and MRS Hive
• Object storage: Object Storage Service (OBS)
• Relational database: RDS for MySQL, RDS for PostgreSQL, RDS for SQL Server, MySQL, PostgreSQL, Microsoft SQL Server, and Oracle
• NoSQL: CloudTable
• Search: Elasticsearch and Cloud Search Service (CSS)</td><td rowspan="3">• Supported by local storage. Only MRS Hive and MRS Hudi are supported in storage-compute decoupling scenarios.
• Only MRS Hive is supported in Ranger scenarios.
• Not supported if SSL is enabled for ZooKeeper
• Recommended MRS HDFS versions:
– 2.8.X
– 3.1.X
• Recommended MRS HBase versions:
– 2.1.X
– 1.3.X
• MRS Hive and MRS Hudi 2.x versions are not supported. The following versions are recommended:
– 1.2.X
– 3.1.X</td></tr>
<tr><td>MRS Hive</td><td>• Data warehouse: GaussDB(DWS), Data Lake Insight (DLI), and MRS ClickHouse
• Hadoop: MRS HDFS, MRS HBase, and MRS Hive
• Object storage: Object Storage Service (OBS)
• Relational database: RDS for MySQL, RDS for PostgreSQL, RDS for SQL Server, MySQL, PostgreSQL, Microsoft SQL Server, and Oracle
• NoSQL: CloudTable
• Search: Elasticsearch and Cloud Search Service (CSS)</td></tr>
<tr><td>MRS Hudi</td><td>Data warehouse: GaussDB(DWS)</td></tr>
</table>

Category	Source	Destination	Description
	FusionInsight HDFS FusionInsight HBase FusionInsight Hive	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object storage: Object Storage Service (OBS) • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	• FusionInsight cannot serve as the destination. • Supported only by local storage and not in storage-compute decoupling scenarios • Not supported by Ranger • Not supported if SSL is enabled for ZooKeeper • Recommended FusionInsight HDFS versions: – 2.8.X – 3.1.X • Recommended FusionInsight HBase versions: – 2.1.X – 1.3.X • Recommended FusionInsight Hive versions: – 1.2.X – 3.1.X

Category	Source	Destination	Description
	Apache HBase Apache Hive Apache HDFS	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object storage: Object Storage Service (OBS) • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	• Apache cannot serve as the destination. • Supported only by local storage and not in storage-compute decoupling scenarios • Not supported by Ranger • Not supported if SSL is enabled for ZooKeeper • Recommended Apache HBase versions: – 2.1.X – 1.3.X • Apache Hive 2.*x* versions are not supported. The following versions are recommended: – 1.2.X – 3.1.X • Recommended Apache HDFS versions: – 2.8.X – 3.1.X
Object storage	Object Storage Service (OBS)	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	• Object Storage Migration Service (OMS) is recommended for migration between object storage services. • Binary files cannot be imported to a database or NoSQL.

Category	Source	Destination	Description
File system	FTP SFTP	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	• The file system cannot serve as the destination. • Only text files such as CSV files can be migrated from FTP or SFTP servers to search services. Binary files cannot. • obsutil is recommended for migrating data from HTTP servers to OBS. For details, see **Introduction to obsutil**.
	HTTP	Hadoop: MRS HDFS	
Relational database	RDS for MySQL	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, MRS Hive, and MRS Hudi • Object storage: Object Storage Service (OBS) • NoSQL: CloudTable • Relational database: RDS for MySQL, RDS for PostgreSQL, and RDS for SQL Server • Search: Elasticsearch and Cloud Search Service (CSS)	• You are advised to use Data Replication Service (DRS) to migrate data between OLTP databases. • RDS for MySQL does not support the SSL mode. • Recommended Microsoft SQL Server version: 2005 or later • The KingBase database and GaussDB can be connected using the PostgreSQL connector. The supported source and destination are the same as those of the PostgreSQL data source.
	RDS for SQL Server RDS for PostgreSQL	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object storage: Object Storage Service (OBS) • NoSQL: CloudTable • Relational database: RDS for MySQL, RDS for PostgreSQL, and RDS for SQL Server • Search: Elasticsearch and Cloud Search Service (CSS)	

Category	Source	Destination	Description
	MySQL PostgreSQL Oracle	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, MRS Hive, and MRS Hudi • Object-based storage: Object Storage Service (OBS) • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	
	Microsoft SQL Server	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object-based storage: Object Storage Service (OBS) • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	

Category	Source	Destination	Description
	SAP HANA	• Data warehouse: Data Lake Insight (DLI) • Hadoop: MRS Hive	SAP HANA data sources have the following restrictions: • SAP HANA cannot serve as the destination. • Only the 2.00.050.00.1592305219 version is supported. • Only the Generic Edition is supported. • BW/4 FOR HANA is not supported. • Only database names, table names, and column names consisting of English letters are supported. Special characters such as spaces and symbols are not allowed. • The following data types are supported: date, digit, Boolean, and character (except SHORTTEXT). Other data types such as binary are not supported. • During migration, tables cannot be automatically created at the destination.

Category	Source	Destination	Description
	Database sharding	• Data warehouse: Data Lake Insight (DLI) • Hadoop: MRS HBase and MRS Hive • Search: Elasticsearch and Cloud Search Service (CSS) • Object-based storage: Object Storage Service (OBS)	Database shards cannot serve as the destination. A shard link connects to multiple backend data sources at the same time. The link can be used as the job source to migrate data from those data sources to other data sources.
NoSQL	Redis	Hadoop: MRS HDFS, MRS HBase, and MRS Hive	NoSQL except CloudTable cannot serve as the destination.
	Document Database Service (DDS)		
	MongoDB		
	CloudTable HBase	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object storage: Object Storage Service (OBS) • Relational database: RDS for MySQL, RDS for PostgreSQL, RDS for SQL Server, MySQL, PostgreSQL, Microsoft SQL Server, and Oracle • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	
	Cassandra	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object storage: Object Storage Service (OBS) • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	

Category	Source	Destination	Description
Message system	Data Ingestion Service (DIS)	Search: Cloud Search Service (CSS)	The message system cannot serve as the destination.
	Apache Kafka		
	DMS Kafka		
	MRS Kafka	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object storage: Object Storage Service (OBS) • Relational database: RDS for MySQL, RDS for PostgreSQL, and RDS for SQL Server • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	• MRS Kafka cannot serve as the destination. • Supported only by local storage and not in storage-compute decoupling scenarios • Not supported by Ranger • Not supported if SSL is enabled for ZooKeeper
Search	Elasticsearch	• Data warehouse: GaussDB(DWS) and Data Lake Insight (DLI) • Hadoop: MRS HDFS, MRS HBase, and MRS Hive • Object storage: Object Storage Service (OBS) • Relational database: RDS for MySQL, RDS for PostgreSQL, and RDS for SQL Server • NoSQL: CloudTable • Search: Elasticsearch and Cloud Search Service (CSS)	Only the non-security mode is supported.
	Cloud Search Service (CSS)		You are advised to use Logstash to import data to CSS. For details, see **Using Logstash to Import Data to Elasticsearch**

NOTE

In the preceding table, the non-cloud data sources, such as MySQL, include on-premises MySQL, MySQL built on ECSs, or MySQL on the third-party cloud.

Supported Data Sources in Entire DB Migration

Entire DB migration is used when an on-premises data center or a database created on an ECS needs to be synchronized to a database service or big data

service on the cloud. It is suitable for offline database migration but not online real-time migration.

Table 2-4 lists the data sources supporting entire DB migration using CDM.

Table 2-4 Supported data sources in entire DB migration

Category	Data Source	Read	Write	Description
Data warehouse	Data Warehouse Service (DWS)	Supported	Supported	-
Hadoop (available only for local storage, and not for storage-compute decoupling, Ranger, or ZooKeeper for which SSL is enabled)	MRS HBase	Supported	Supported	Entire DB migration only to MRS HBase Recommended versions: • 2.1.X • 1.3.X
	MRS Hive	Supported	Supported	Entire DB migration only to a relational database 2.*x* versions are not supported. The following versions are recommended: • 1.2.X • 3.1.X
	FusionInsight HBase	Supported	Not supported	Recommended versions: • 2.1.X • 1.3.X
	FusionInsight Hive	Supported	Not supported	Entire DB migration only to a relational database 2.*x* versions are not supported. The following versions are recommended: • 1.2.X • 3.1.X

Category	Data Source	Read	Write	Description
	Apache HBase	Supported	Not supported	Recommended versions: • 2.1.X • 1.3.X
	Apache Hive	Supported	Not supported	Entire DB migration only to a relational database 2.*x* versions are not supported. The following versions are recommended: • 1.2.X • 3.1.X
Relational database	RDS for MySQL	Supported	Supported	Migration from OLTP to OLTP is not supported. In this scenario, you are advised to use the Data Replication Service (DRS).
	RDS for PostgreSQL	Supported	Supported	
	RDS for SQL Server	Supported	Supported	
	MySQL	Supported	Not supported	
	PostgreSQL	Supported	Not supported	
	Microsoft SQL Server	Supported	Not supported	
	Oracle	Supported	Not supported	

Category	Data Source	Read	Write	Description
	SAP HANA	Supported	Not supported	• Only the 2.00.050.00.1592305219 version is supported. • Only the Generic Edition is supported. • BW/4 FOR HANA is not supported. • Only database names, table names, and column names consisting of English letters are supported. Special characters such as spaces and symbols are not allowed. • The following data types are supported: date, digit, Boolean, and character (except SHORTTEXT). Other data types such as binary are not supported. • During migration, tables cannot be automatically created at the destination.
	Dameng database	Supported	Not supported	Only to DWS and Hive

Category	Data Source	Read	Write	Description
NoSQL	Redis	Supported	Supported	-
	Document Database Service (DDS)	Supported	Supported	Only migration between DDS and MRS is supported.
	CloudTable Service (CloudTable)	Supported	Supported	-

2.3 Supported Data Types

To ensure that data is completely imported to the migration destination, correctly configure field mappings based on data types supported for different data sources. For details, see **Table 2-5**.

Table 2-5 Supported data types

Data Connection Type	Data Type
MySQL	**Data Types Supported in MySQL Database Migration**
SQL Server	**Data Types Supported in SQL Server Database Migration**
Oracle	**Data Types Supported in Oracle Database Migration**
PostgreSQL	**Data Types Supported in PostgreSQL Database Migration**
ShenTong	**Data Types Supported in ShenTong Database Migration**
SAP HANA	**Data Types Supported in SAP HANA Database Migration**
DWS	**Data Types Supported in DWS Database Migration**
Dameng	**Data Types Supported in Dameng Database Migration**
DLI	**Data Types Supported in DLI Database Migration**
Elasticsearch/Cloud Search Service (CSS)	**Data Types Supported in Elasticsearch/CSS Database Migration**

Data Types Supported in MySQL Database Migration

When the source end is a MySQL database and the destination end is a Hive or DWS database, the following data types are supported:

Table 2-6 Data types supported for the open-source MySQL database

Category	Type	Description	Storage Format Example	Hive	DWS
Character string	CHAR(M)	A fixed-length string of 1 to 255 characters, for example, CHAR(5). The length limit is not mandatory. It is set to 1 by default.	'a' or 'aaaaa'	CHAR	CHAR
	VARCHAR(M)	A variable-length string consists of 1 to 255 characters (more than 255 characters for MySQL of a later version). Example: VARCHAR(25). When creating a field of the VARCHAR type, you must define the length.	'a' or 'aaaaa'	VARCHAR	VARCHAR
Value	DECIMAL(M, D)	Uncompressed floating-point numbers cannot be unsigned. In unpacking decimals, each decimal corresponds to a byte. Defining the number of display lengths (M) and decimals (D) is required. NUMERIC is the synonym of DECIMAL.	52.36	DECIMAL	When D is 0, it corresponds to BIGINT. When D is not 0, it corresponds to NUMBERIC.
	NUMBERIC	Same as DECIMAL	-	DECIMAL	NUMBERIC

Category	Type	Description	Storage Format Example	Hive	DWS
	INTEGER	An integer of normal size that can be signed. If the value is signed, it ranges from -2147483648 to 2147483647. If the value is unsigned, the value ranges from 0 to 4294967295. Up to 11-bit width can be specified.	5236	INT	INTEGER
	INTEGER UNSIGNED	Unsigned form of INTEGER	-	BIGINT	INTEGER
	INT	Same as INTEGER	5236	INT	INTEGER
	INT UNSIGNED	Same as INTEGER UNSIGNED	-	BIGINT	INTEGER
	BIGINT	A large integer that can be signed. If the value is signed, it ranges from -9223372036854775808 to 9223372036854775807. If the value is unsigned, the value ranges from 0 to 18446744073709551615. Up to 20-bit width can be specified.	5236	BIGINT	BIGINT
	BIGINT UNSIGNED	Unsigned form of BIGINT	-	BIGINT	BIGINT

Category	Type	Description	Storage Format Example	Hive	DWS
	MEDIUMINT	A medium-sized integer that can be signed. If the value is signed, it ranges from -8388608 to 8388607. If the value is unsigned, it ranges from 0 to 16777215, and you can specify a maximum of 9-bit width.	-128, 127	INT	INTEGER
	MEDIUMINT UNSIGNED	Unsigned form of MEDIUMINT	-	BIGINT	INTEGER
	TINYINT	A very small integer that can be signed. If signed, the value ranges from -128 to 127. If unsigned, the value ranges from 0 to 255, and you can specify a maximum of 4-bit width.	100	TINYINT	SMALLINT
	TINYINT UNSIGNED	Unsigned form of TINYINT	-	TINYINT	SMALLINT
	BOOL	The bool of MySQL is tinyint(1).	-128, 127	SMALLINT	BYTEA
	SMALLINT	A small integer that can be signed. If the value is signed, it ranges from -32768 to 32767. If unsigned, the value ranges from 0 to 65535, and you can specify a maximum of 5-bit width.	9999	SMALLINT	SMALLINT

Category	Type	Description	Storage Format Example	Hive	DWS
	SMALLINT UNSIGNED	Unsigned form of SMALLINT	-	INT	SMALLINT
	REAL	Same as DOUBLE	-	DOUBLE	-
	FLOAT(M,D)	Unsigned floating-point numbers cannot be used. The display length (M) and number of decimal places (D) can be specified. This is not mandatory, and the default value is 10,2. In the preceding information, 2 indicates the number of decimal places and 10 indicates the total number of digits (including decimal places). The decimal precision can reach 24 floating points.	52.36	FLOAT	FLOAT4
	DOUBLE(M,D)	Unsigned double-precision floating-point numbers cannot be used. The display length (M) and number of decimal places (D) can be specified. This is not mandatory. The default value is 16,4, where 4 is the number of decimal places. The decimal precision can reach 53-digit. REAL is a synonym of DOUBLE.	52.36	DOUBLE	FLOAT8
	DOUBLE PRECISION	Similar to DOUBLE	52.3	DOUBLE	FLOAT8

Category	Type	Description	Storage Format Example	Hive	DWS
Bit	BIT(M)	Stored bit type value. BIT(M) can store up to *M* bits of values, and *M* ranges from 1 to 64.	B'1111100' B'1100'	TINYINT	BYTEA
Time and date	DATE	The value is in the *YYYY-MM-DD* format and ranges from **1000-01-01** to **9999-12-31**. For example, **December 30, 1973** will be stored as **1973-12-30**.	1999-10-01	DATE	TIMESTAMP
	TIME	Stores information about the hour, minute, and second.	'09:10:21' or '9:10:21'	Not supported (string)	TIME
	DATETIME	The date and time are in the *YYYY-MM-DD HH:MM:SS* format and range from **1000-01-01 00:00:00** to **9999-12-31 23:59:59**. For example, **3:30 p.m. on December 30, 1973** will be stored as **1973-12-30 15:30:00**.	'1973-12-30 15:30:00'	TIMESTAMP	TIMESTAMP
	TIMESTAMP	Timestamp type. Timestamp between midnight on January 1, 1970 and a time point in 2037. Similar to the DATETIME format (YYYYMMDDHHMMSS), except that no hyphen is required. For example, **3:30 p.m. December 30, 1973** will be stored as **19731230153000**.	19731230153000	TIMESTAMP	TIMESTAMP

Category	Type	Description	Storage Format Example	Hive	DWS
	YEAR(M)	The year is stored in 2-digit or 4-digit number format. If the length is specified as 2 (for example, YEAR(2)), the year ranges from 1970 to 2069 (70 to 69). If the length is specified as 4, the year ranges from 1901 to 2155. The default length is 4.	2000	Not supported (string)	Not supported
Multimedia (binary)	BINARY(M)	The number of bytes is *M*. The length of a variable-length binary string ranges from 0 to *M*. *M* is the value length plus 1.	0x2A3B4058 (binary data)	Not supported	BYTEA
	VARBINARY(M)	The number of bytes is *M*. A fixed binary string with a length of 0 to *M*.	0x2A3B4059 (binary data)	Not supported	BYTEA
	TEXT	The maximum length of the field is 65535 characters. TEXT is a "binary large object" and is used to store large binary data, such as images or other types of files.	0x5236 (binary data)	Not supported	Not supported
	TINYTEXT	A binary string of 0 to 255 bytes in short text	-	-	Not supported
	MEDIUMTEXT	A binary string of 0 to 167772154 bytes in medium-length text	-	-	Not supported
	LONGTEXT	A binary string of 0 to 4294967295 bytes in large-length text	-	-	Not supported

Category	Type	Description	Storage Format Example	Hive	DWS
	BLOB	The maximum length of the field is 65535 characters. BLOB is a "binary large object" and is used to store large binary data, such as images or other types of files. BLOB is case-sensitive.	0x5236 (binary data)	Not supported	Not supported
	TINYBLOB	A binary string of 0 to 255 bytes in short text	-	Not supported	Not supported
	MEDIUMBLOB	A binary string of 0 to 167772154 bytes in medium-length text	-	Not supported	Not supported
	LONGBLOB	A binary string of 0 to 4294967295 bytes in large-length text	0x5236 (binary data)	Not supported	Not supported
Special type	SET	SET is a string object that can have no or multiple values. The values come from the allowed column of values specified when the table is created. When specifying the SET column values that contain multiple SET members, separate the members with commas (,). The SET member value cannot contain commas (,).	-	-	Not supported
	JSON	-	-	Not supported	Not supported (TEXT)

Category	Type	Description	Storage Format Example	Hive	DWS
	ENUM	When an ENUM is defined, a list of its values is created, which are the items that must be used for selection (or NULL). For example, if you want a field to contain "A", "B", or "C", you can define an ENUM ("A", "B", or "C"). Only these values (or NULL) can be used to fill in the field.	-	Not supported	Not supported

Data Types Supported in Oracle Database Migration

When the source end is an Oracle database and the destination end is a Hive or DWS database, the following data sources are supported:

Table 2-7 Data types supported for the Oracle database

Category	Type	Description	Hive	DWS
Character string	char	Fixed-length character string, which is padded with spaces to reach the maximum length.	CHAR	CHAR
	nchar	Fixed-length character string contains data in Unicode format.	CHAR	CHAR
	varchar2	Synonym of VARCHAR. It is a variable-length string, unlike the CHAR type, which does not pad the field or variable to reach its maximum length with spaces.	VARCHAR	VARCHAR
	nvarchar2	Variable-length character string contains data in Unicode format.	VARCHAR	VARCHAR
Value	number	Stores numbers with a precision of up to 38 digits.	DECIMAL	NUMERIC

Category	Type	Description	Hive	DWS
	binary_float	2-bit single-precision floating point number	FLOAT	FLOAT8
	binary_double	64-bit double-precision floating point number	DOUBLE	FLOAT8
	long	A maximum of 2 GB character data can be stored.	Not supported	Not supported
Time and date	date	7-byte date/time data type, including seven attributes: century, year in the century, month, day in the month, hour, minute, and second.	DATE	TIMESTAMP
	timestamp	7-byte or 11-byte fixed-width date/time data type that contains decimals (seconds)	TIMESTAMP	TIMESTAMP
	timestamp with time zone	3-byte timestamp, which supports the time zone.	TIMESTAMP	TIME WITH TIME ZONE
	timestamp with local time zone	7-byte or 11-byte fixed-width date/time data type. Time zone conversion occurs when data is inserted or read.	TIMESTAMP	Not supported (TEXT)
	interval year to month	5-byte fixed-width data type, which is used to store a time segment.	Not supported	Not supported (TEXT)
	interval day to second	11-byte fixed-width data type, which is used to store a time segment. The time segment is stored in days/hours/minutes/seconds. The value can also contain nine decimal places (seconds).	Not supported	Not supported (TEXT)
Multimedia (binary)	raw	A variable-length binary data type. Character set conversion is not performed for data stored in this data type.	Not supported	Not supported

Category	Type	Description	Hive	DWS
	long raw	Stores up to 2 GB binary information.	Not supported	Not supported
	blob	A maximum of 4 GB data can be stored.	Not supported	Not supported
	clob	In Oracle 10g and later versions, a maximum of (4 GB) x (database block size) bytes of data can be stored. CLOB contains the information for which character set conversion is to be performed. This data type is ideal for storing plain text information.	String	Not supported
	nclob	This type can store a maximum of 4 GB data. When the character set is converted, this type is affected.	Not supported	Not supported
	bfile	An Oracle directory object and a file name can be stored in the database column, and the file can be read through the Oracle directory object and file name.	Not supported	Not supported
Others	rowid	It is the address of a row in the database table. It is 10 bytes long.	Not supported	Not supported
	urowid	It is a common row ID and does not have a fixed rowid table.	Not supported	Not supported

Data Types Supported in SQL Server Database Migration

When the source end is a SQL Server database and the destination end is a Hive, Oracle or DWS database, the following data sources are supported:

Table 2-8 Data types supported for the SQL Server database

Category	Type	Description	Hive	DWS	Oracle
String data type	char	Fixed-length character string, which is padded with spaces to reach the maximum length.	CHAR	CHAR	CHAR

Category	Type	Description	Hive	DWS	Oracle
	nchar	Fixed-length character string contains data in Unicode format.	CHAR	CHAR	CHAR
	varchar	A variable-length string consists of 1 to 255 characters (more than 255 characters for MySQL of a later version). Example: VARCHAR(25). When creating a field of the VARCHAR type, you must define the length.	VARCHAR	VARCHAR	VARCHAR
	nvarchar	Stores variable-length Unicode character data, similar to varchar.	VARCHAR	VARCHAR	VARCHAR
Numeric data type	int	int is stored in four bytes, where one binary bit represents a sign bit, and the other 31 binary bits represent a length and a size, and may represent all integers ranging from -2^{31} to $2^{31} - 1$.	INT	INTEGER	INT
	bigint	bigint is stored in eight bytes, where one binary bit represents a sign bit, and the other 63 binary bits represent a length and a size, and may represent all integers ranging from -2^{63} to $2^{63} - 1$.	BIGINT	BIGINT	NUMBER
	smallint	Data of the smallint type occupies two bytes of storage space. One binary bit indicates a positive or negative sign of an integer value, and the other 15 binary bits indicate a length and a size, and may represent all integers ranging from -2^{15} to 2^{15}.	SMALLINT	SMALLINT	NUMBER
	tinyint	Tinyint data occupies one byte of storage space and can represent all integers ranging from 0 to 255.	TINYINT	TINYINT	NUMBER
	real	The value can be a positive or negative decimal number.	DOUBLE	FLOAT4	NUMBER
	float	The number of digits (in scientific notation) of the mantissa of a float value, which determines the precision and storage size	FLOAT	FLOAT8	binary_float
	decimal	Numeric data type with fixed precision and scale	DECIMAL	NUMERIC	NUMBER

Category	Type	Description	Hive	DWS	Oracle
	numeric	Stores zero, positive, and negative fixed point numbers.	DECIMAL	NUMERIC	NUMBER
Date and time data type	date	Stores date data represented by strings.	DATE	TIMESTAMP	DATE
	time	Time of a day, which is recorded in the form of a character string.	Not supported (string)	TIME	Not supported
	datetime	Stores time and date data.	TIMESTAMP	TIMESTAMP	Not supported
	datetime2	Extended type of datetime, which has a larger data range. By default, the minimum precision is the highest, and the user-defined precision is optional.	TIMESTAMP	TIMESTAMP	Not supported
	smalldatetime	The smalldatetime type is similar to the datetime type. The difference is that the smalldatetime type stores data from January 1, 1900 to June 6, 2079. When the date and time precision is low, the smalldatetime type can be used. Data of this type occupies 4-byte storage space.	TIMESTAMP	TIMESTAMP	Not supported
	datetimeoffset	A time that uses the 24-hour clock and combined with date and the time zone.	Not supported (string)	TIMESTAMP	Not supported
Multimedia data types (binary)	text	Stores text data.	Not supported (string)	Not supported (string)	Not supported
	netxt	The function of this type is the same as that of the text type. It is non-Unicode data with variable length.	Not supported (string)	Not supported (string)	Not supported

Category	Type	Description	Hive	DWS	Oracle
	image	Variable-length binary data used to store pictures, catalog pictures, or paintings.	Not supported (string)	Not supported (string)	Not supported
	binary	Binary data with a fixed length of *n* bytes, where *n* ranges from 1 to 8,000.	Not supported (string)	Not supported (string)	Not supported
	varbinary	Variable-length binary data	Not supported (string)	Not supported (string)	Not supported
Currency data type	money	Stores currency values.	Not supported (string)	Not supported (string)	Not supported
	small money	Similar to the money type, a currency symbol is prefixed to the input data. For example, the currency symbol of CNY is ¥.	Not supported (string)	Not supported (string)	Not supported
Data type	bit	Bit data type. The value is 0 or 1. The length is 1 byte. A bit value is often used as a logical value to determine whether it is true(1) or false(0). If a non-zero value is entered, the system replaces it with 1.	Not supported	Not supported	Not supported
Other data types	rowversion	Each piece of data has a counter. The value of the counter increases when an insert or update operation is performed on a table that contains the **rowversion** column in the database.	Not supported	Not supported	Not supported

Category	Type	Description	Hive	DWS	Oracle
	unique identifier	A 16-byte globally unique identifier (GUID) is a unique number generated by the SQL Server based on the network adapter address and host CPU clock. Each GUID is a hexadecimal number ranging from 0 to 9 or a to f.	Not supported	Not supported	Not supported
	cursor	Cursor data type	Not supported	Not supported	Not supported
	sql_variant	Stores any valid SQL Server data except the text, image, and timestamp data, which facilitates the development of the SQL Server.	Not supported	Not supported	Not supported
	table	Stores the result set after a table or view is processed.	Not supported	Not supported	Not supported
	xml	Data type of the XML data. XML instances can be stored in columns or variables of the XML type. The stored XML instance size cannot exceed 2 GB.	Not supported	Not supported	Not supported

Data Types Supported in PostgreSQL Database Migration

When the source end is a PostgreSQL database and the destination end is Hive, DLI, or DWS, the following data types are supported:

Table 2-9 Data types supported for the PostgreSQL database

Category	Type	Description	Hive	DWS	DLI
Character	char	Fixed-length string, which is padded to a specified length with spaces on the right.	CHAR	CHAR	Not supported (string)

Category	Type	Description	Hive	DWS	DLI
	varchar	Variable-length string. Fields or variables are not padded to the maximum length with spaces.	CARCHAR	CARCHAR	Not supported (string)
Value	smallint	The extension name int2 is stored in two bytes and ranges from –32768 to 32767.	SMALLINT	SMALLINT	SMALLINT
	int	The extension name int4 is stored in four bytes and ranges from –2147483648 to 2147483647.	INTEGER	INT	INT
	bigint	The extension name int8 is stored in eight bytes and ranges from –9223372036854775808 to 9223372036854775807.	BIGINT	BIGINT	BIGINT
	decimal(p,s)	The precision p represents the number of valid digits stored in the value, and the scale s represents the number of digits after the decimal point that can be stored. The maximum value of p is 1000.	DECIMAL(P,S)	DECIMAL(P,S)	DECIMAL(P,S)

Category	Type	Description	Hive	DWS	DLI
	float	4-byte or 8-byte storage. float(n): For the single precision, the value of n ranges from 1 to 24, the number of valid precision digits is 6, and the length is four bytes. For the double precision, the value of n ranges from 25 to 53, the number of valid precision digits is 15, and the length is 8 bytes.	FLOAT/ DOUBLE	FLOAT/ DOUBLE	FLOAT/ DOUBLE
	smallserial	Sequence data type, which is stored in smallint format	SMALLINT	SMALLINT	SMALLINT
	serial	Sequence data type, which is stored in int format	INTEGER	INT	INT
	bigserial	Sequence data type, which is stored in bigint format	BIGINT	BIGINT	BIGINT
Time and date	date	Stores the date.	DATE	DATE	DATE
	timestamp	Stores date and time data without time zones.	TIMESTAMP	TIMESTAMP	Not supported (string)
	timestamptz	Stores the date and time, including the time zone.	TIMESTAMP	TIMESTAMPZ	Not supported (string)
	time	Time within one day, excluding the time zone	Not supported (string)	TIME	Not supported (string)
	timez	Time within one day, including the time zone	Not supported (string)	TIMEZ	Not supported (string)

Category	Type	Description	Hive	DWS	DLI
	interval	Time interval	Not supported (string)	Not supported (string)	Not supported (string)
Bit string	bit	Fixed-length string, for example, **b'000101'**	Not supported (string)	Not supported (string)	Not supported (string)
	varbit	Variable-length string, for example, **b'101'**	Not supported (string)	Not supported (string)	Not supported (string)
Currency type	money	The value is stored in eight bytes and ranges from –922337203685477.5808 to 922337203685477.5807.	DOUBLE	MONEY	DECIMAL(P,S)
Boolean	boolean	The value is stored in one byte and can be **1**, **0**, or **NULL**.	BOOLEAN	BOOLEAN	BOOLEAN
Text type	text	Variable-length text without a length limit	Not supported (string)	Not supported (string)	Not supported (string)

Data Types Supported in DWS Database Migration

If the migration source is a DWS database, the following data types are supported.

Table 2-10 Data types supported for the DWS database

Category	Type	Description
Character	char	Fixed-length string, which is padded to a specified length with spaces on the right.
	varchar	Variable-length string. Fields or variables are not padded to the maximum length with spaces.
Value	double	Stores double-precision floating-point numbers.

Category	Type	Description
	decimal(p,s)	The precision p represents the number of valid digits stored in the value, and the scale s represents the number of digits after the decimal point that can be stored. The maximum value of p is 1000.
	numeric	Stores zero, positive, and negative fixed point numbers.
	real	Same as double
	int	int is stored in four bytes, where one binary bit represents a sign bit, and the other 31 binary bits represent a length and a size, and may represent all integers ranging from -2^{31} to $2^{31} - 1$.
	bigint	bigint is stored in eight bytes, where one binary bit represents a sign bit, and the other 63 binary bits represent a length and a size, and may represent all integers ranging from -2^{63} to $2^{63} - 1$.
	smallint	Data of the smallint type occupies two bytes of storage space. One binary bit indicates a positive or negative sign of an integer value, and the other 15 binary bits indicate a length and a size, and may represent all integers ranging from -2^{15} to 2^{15}.
	tinyint	Tinyint data occupies one byte of storage space and can represent all integers ranging from 0 to 255.
Time and date	date	Stores the date.
	timestamp	Stores date and time data without time zones.
	time	Time within one day, excluding the time zone
Bit string	bit	Fixed-length string, for example, **b'000101'**
Boolean	boolean	The value is stored in one byte and can be **1**, **0**, or **NULL**.
Text type	text	Variable-length text without a length limit

Data Types Supported in ShenTong Database Migration

When the source is a ShenTong database and the destination is MRS Hive or MRS Hudi, the following data types are supported.

Table 2-11 Data types supported for the ShenTong database

Category	Type	Description	Storage Format Example	MRS Hive	MRS Hudi
Character	VARCHAR	Stores specified fixed-length character strings.	'a' or 'aaaaa'	VARCHAR(765)	STRING
	BPCHAR	Stores specified variable-length character strings.	'a' or 'aaaaa'	VARCHAR(765)	STRING
Value	NUMERIC	Stores zero, positive, and negative fixed point numbers.	52.36	DECIMAL(10,0)	DECIMAL(18,0)
	INT	Stores zero, positive, and negative fixed point numbers.	5236	INT	INT
	BIGINT	Stores signed integers. Integer part: 19 digits; decimal part: 0 digits	5236	BIGINT	BIGINT
	TINYINT	Stores signed integers. Integer part: 3 digits; decimal part: 0 digits	100	SMALLINT	INT
	BINARY	Stores fixed-length binary data.	0x2A3B4058	Not supported	FLOAT
	VARBINARY	Stores variable-length binary data.	0x2A3B4058	Not supported	BINARY
	FLOAT	Stores floating-point numbers with binary precision.	52.36	FLOAT	FLOAT
	DOUBLE	Stores double-precision floating-point numbers.	52.3	DOUBLE	DOUBLE

Category	Type	Description	Storage Format Example	MRS Hive	MRS Hudi
Time and date	DATE	Stores information about the year, month, and day.	'1999-10-01', '1999/10/01', or '1999.10.01'	DATE	DATE
	TIME	Stores information about the hour, minute, and second.	'09:10:21' or '9:10:21'	STRING	STRING
	TIMESTAMP	Stores information about the year, month, day, hour, minute, and second.	2002-12-12 09:10:21','2002-12-12 9:10:21' '2002/12/12 09:10:21' or '2002.12.12 09:10:21'	TIMESTAMP	TIMESTAMP
Multimedia	CLOB	Stores variable-length binary large objects with a maximum length of 2 GB minus 1 byte.	0x5236 (binary data)	STRING	STRING
	BLOB	Stores variable-length binary large objects with a maximum length of 2 GB minus 1 byte.	0x5236 (binary data)	Not supported	BINARY
Boolean	BOOLEAN	The value is stored in one byte and can be **1**, **0**, or **NULL**.	1	BOOLEAN	BOOLEAN

Data Types Supported in SAP HANA Database Migration

If the source is an SAP HANA database, the following data types are supported.

Table 2-12 Data types supported for the SAP HANA database

Category	Type	Description
Character	VARCHAR	Stores specified fixed-length character strings.
	NVARCHAR	Variable-length character string contains data in Unicode format.
	TEXT	It is used to store long character strings. The maximum length of a string is 2 GB minus 1 byte. Long text strings are stored.
Value	BIGINT	Stores signed integers. Integer part: 19 digits; decimal part: 0 digits
	TINYINT	Stores signed integers. Integer part: 3 digits; decimal part: 0 digits
	SMALLINT	Data of the smallint type occupies two bytes of storage space. One binary bit indicates a positive or negative sign of an integer value, and the other 15 binary bits indicate a length and a size, and may represent all integers ranging from -2^{15} to 2^{15}.
	REAL	The value can be a positive or negative decimal number.
	DECIMAL	Numeric data type with fixed precision and scale
	FLOAT	Stores floating-point numbers with binary precision.
	DOUBLE	Stores double-precision floating-point numbers.
Time and date	DATE	Stores information about the year, month, and day.
	TIME	Stores information about the hour, minute, and second.
	TIMESTAMP	Stores information about the year, month, day, hour, minute, and second.
Multimedia	CLOB	Stores variable-length binary large objects with a maximum length of 2 GB minus 1 byte.
	NCLOB	This type can store a maximum of 4 GB data. When the character set is converted, this type is affected.
Boolean	BOOLEAN	The value is stored in one byte and can be **1**, **0**, or **NULL**.

Data Types Supported in DLI Database Migration

If the migration source is a DLI database, the following data types are supported.

Table 2-13 Data types supported for the DLI database

Category	Type	Description
Character	CHAR	Stores specified fixed-length character strings.
	VARCHAR	Same as CHAR
	STRING	It is used to store long character strings. The maximum length of a character string is 2 GB minus 1 byte. Long text strings are stored.
Value	BIGINT	Stores signed integers. Integer part: 19 digits; decimal part: 0 digits
	TINYINT	Stores signed integers. Integer part: 3 digits; decimal part: 0 digits
	SMALLINT	Data of the smallint type occupies two bytes of storage space. One binary bit indicates a positive or negative sign of an integer value, and the other 15 binary bits indicate a length and a size, and may represent all integers ranging from -2^{15} to 2^{15}.
	INT	Stores signed integers. Integer part: 10 digits; decimal part: 0 digits
	DECIMAL	Numeric data type with fixed precision and scale
	FLOAT	Stores floating-point numbers with binary precision.
	DOUBLE	Stores double-precision floating-point numbers.
Time and date	DATE	Stores information about the year, month, and day.
	TIMESTAMP	Stores information about the year, month, day, hour, minute, and second.
Boolean	BOOLEAN	The value is stored in one byte and can be **1**, **0**, or **NULL**.

Data Types Supported in Elasticsearch/CSS Database Migration

If the migration source is an Elasticsearch/CSS database, the following data types are supported.

Table 2-14 Data types supported for the Elasticsearch/CSS database

Category	Type	Description	Storage Format Example	MySQL
Character	keyword	Stores character strings.	"keyword"	String

Category	Type	Description	Storage Format Example	MySQL
	text	Stores long character strings. The maximum length of a character string is 2 GB minus 1 byte. Long text strings are stored.	"long string"	TEXT
	string	Stores long character strings. The maximum length of a character string is 2 GB minus 1 byte. Long text strings are stored.	"a string"	String
Integer	short	Stores 16-bit signed integers ranging from –32768 to 32767.	32765	smallInt
	integer	Stores 32-bit signed integers ranging from -2^{31} to $2^{31} - 1$.	3276566	int
	long	Stores 64-bit signed integers ranging from -2^{63} to $2^{63} - 1$.	3276566666	BIGINT
Value	double	64-bit IEEE 754 double-precision floating-point format	21.333	double
	float	32-bit IEEE 754 single-precision floating-point format	21.333	double
Boolean	boolean	The value is stored in one byte and can be **1**, **0**, or **NULL**.	1	Boolean
Object	object	A string of flat storage objects	{"users.name": ["John","Smith"], users.age": [26,28], "users.sex" :[1,2]}	TEXT

Category	Type	Description	Storage Format Example	MySQL
Nested	nested	A string of nested storage objects	{"users.name" : "John" , "users.age" : 26, "users.sex" : 1} { "users.name" : "Smith", "users.age" : 28, "users.sex" : 2}	TEXT
Date	date	A string in the date format	"2018-01-13" or "2018-01-13 12:10:30"	DATE or timeStamp
Special type	ip	A string in the IP address format	"192.168.127.100"	String
Array	string_array	An array of strings	["str","str"]	TEXT
	short_array	An array of 16-bit integers	[1,1,1]	TEXT
	integer_array	An array of 32-bit integers	[1,1,1]	TEXT
	long_array	An array of 64-bit integers	[1,1,1]	TEXT
	float_array	An array of 32-bit floating-point numbers	[1.0,1.0,1.0]	TEXT
	double_array	An array of 64-bit floating-point numbers	[1.0,1.0,1.0]	TEXT
Value range	completion	A string that is automatically completed	"string"	TEXT

Data Types Supported in Dameng Database Migration

When the source end is a Dameng database and the destination end is a Hive or DWS database, the following data types are supported.

Table 2-15 Data types supported for the Dameng database

Category	Type	Description	Storage Format Example	Hive	DWS
Character	CHAR	Stores specified fixed-length character strings.	'a' or 'aaaaa'	CHAR	CHAR
	CHARACTER	Same as CHAR	'a' or 'aaaaa'	CHAR	CHAR
	VARCHAR	Stores specified variable-length character strings.	'a' or 'aaaaa'	VARCHAR	VARCHAR
	VARCHAR2	Same as VARCHAR	'a' or 'aaaaa'	VARCHAR	VARCHAR
Value	NUMERIC	Stores zero, positive, and negative fixed point numbers.	52.36	DECIMAL	NUMERIC
	DECIMAL	Similar to NUMERIC	52.36	DECIMAL	NUMERIC
	DEC	Same as DECIMAL	52.36	DECIMAL	NUMERIC
	NUMBER	Same as NUMERIC	52.36	DECIMAL	NUMERIC
	INTEGER	Stores signed integers. Integer part: 10 digits; decimal part: 0 digits	5236	INT	INTEGER
	INT	Same as INTEGER	5236	INT	INTEGER
	BIGINT	Stores signed integers. Integer part: 19 digits; decimal part: 0 digits	5236	BIGINT	BIGINT

Category	Type	Description	Storage Format Example	Hive	DWS
	TINYINT	Stores signed integers. Integer part: 3 digits; decimal part: 0 digits	100	TINYINT	SMALLINT
	SMALLINT	Stores signed integers. Integer part: 5 digits; decimal part: 0 digits	9999	SMALLINT	SMALLINT
	BYTE	Similar to TINYINT. Integer part: 3 digits; decimal part: 0 digits	100	TINYINT	SMALLINT
	BINARY	Stores fixed-length binary data.	0x2A3B4058	BINARY (NULL)	BYTEA (NULL)
	VARBINARY	Stores variable-length binary data.	0x2A3B4058	BINARY (NULL)	BYTEA (NULL)
	FLOAT	Stores floating-point numbers with binary precision.	52.36	FLOAT	FLOAT8
	DOUBLE	Similar to FLOAT	52.36	DOUBLE	FLOAT8
	REAL	Stores binary floating-point numbers.	52.3	FLOAT	FLOAT4
	DOUBLE PRECISION	Stores double-precision floating-point numbers.	52.3	DOUBLE	FLOAT8
Bit string	BIT	Stores 1, 0, or NULL.	1, 0, or NULL	TINYINT(1 0 NULL)	BOOLEAN(true false NULL)
Time and date	DATE	Stores information about the year, month, and day.	'1999-10-01', '1999/10/01', or '1999.10.01'	DATE	TIMESTAMP

Category	Type	Description	Storage Format Example	Hive	DWS
	TIME	Stores information about the hour, minute, and second.	'09:10:21' or '9:10:21'	Not supported (string)	TIME
	TIMESTAMP	Stores information about the year, month, day, hour, minute, and second.	2002-12-12 09:10:21','2002-12-12 9:10:21' '2002/12/12 09:10:21' or '2002.12.12 09:10:21'	TIMESTAMP	TIMESTAMP
	TIME WITH TIME ZONE	Stores a TIME value with a time zone. Add the time zone information to the end of the TIME type.	'09:10:21 +8:00', '09:10:21+8:00' or '9:10:21+8:00'	Not supported (string)	TIME WITH TIME ZONE
	TIMESTAMP WITH TIME ZONE	Stores a TIMESTAMP value with a time zone. Add the time zone information to the end of the TIMESTAMP type.	2002-12-12 09:10:21 +8:00','2002-12-12 9:10:21 +8:00' '2002/12/12 09:10:21 +8:00' or '2002.12.12 09:10:21 +8:00'	TIMESTAMP	TIMESTAMP WITH TIME ZONE
	TIMESTAMP WITH LOCAL TIME ZONE	Stores the TIMESTAMP value of a local time zone. The standard time zone type (TIMESTAMP WITH TIME ZONE) can be converted to the local time zone type.	2002-12-12 09:10:21 +8:00','2002-12-12 9:10:21 +8:00' '2002/12/12 09:10:21 +8:00' or '2002.12.12 09:10:21 +8:00'	Not supported (string)	Not supported (TEXT)

Category	Type	Description	Storage Format Example	Hive	DWS
	DATETIME WITH TIME ZONE	Same as TIMESTAMP WITH TIME ZONE	2002-12-12 09:10:21 +8:00','2002-12-12 9:10:21 +8:00' '2002/12/12 09:10:21 +8:00' or '2002.12.12 09:10:21 +8:00'	TIMESTAMP	TIMESTAMP WITH TIME ZONE
	INTERVAL YEAR	Interval of years. The leading precision specifies the range of years.	INTERVAL '0015' YEAR	Not supported (string)	Not supported (VARCHAR)
	INTERVAL YEAR TO MONTH	Interval of months and years. The leading precision specifies the range of years.	INTERVAL '0015-08' YEAR TO MONTH	Not supported (string)	Not supported (VARCHAR)
	INTERVAL MONTH	Interval of months. The leading precision specifies the range of months.	INTERVAL '0015' MONTH	Not supported (string)	Not supported (VARCHAR)
	INTERVAL DAY	Interval of days. The leading precision specifies the range of days.	INTERVAL '150' DAY	Not supported (string)	Not supported (VARCHAR)
	INTERVAL DAY TO HOUR	Interval of hours and days. The leading precision specifies the range of days.	INTERVAL '9 23' DAY TO HOUR	Not supported (string)	Not supported (VARCHAR)
	INTERVAL DAY TO MINUTE	Interval of minutes, hours, and days. The leading precision specifies the range of days.	INTERVAL '09 23:12' DAY TO MINUTE	Not supported (string)	Not supported (VARCHAR)

Category	Type	Description	Storage Format Example	Hive	DWS
	INTERVAL DAY TO SECOND	Interval of seconds, minutes, hours, and days. The leading precision specifies the range of days.	INTERVAL '09 23:12:01.1' DAY TO SECOND	Not supported (string)	Not supported (VARCHAR)
	INTERVAL HOUR	Interval of hours. The leading precision specifies the range of hours.	INTERVAL '150' HOUR	Not supported (string)	Not supported (VARCHAR)
	INTERVAL HOUR TO MINUTE	Interval of minutes and hours. The leading precision specifies the range of hours.	INTERVAL '23:12' HOUR TO MINUTE	Not supported (string)	Not supported (VARCHAR)
	INTERVAL HOUR TO SECOND	Interval of seconds, minutes, and hours. The leading precision specifies the range of hours.	INTERVAL '23:12:01.1' HOUR TO SECOND	Not supported (string)	Not supported (VARCHAR)
	INTERVAL MINUTE	Interval of minutes. The leading precision specifies the range of minutes.	INTERVAL '150' MINUTE	Not supported (string)	Not supported (VARCHAR)
	INTERVAL MINUTE TO SECOND	Interval of seconds and minutes. The leading precision specifies the range of minutes .	INTERVAL '12:01.1' MINUTE TO SECOND	Not supported (string)	Not supported (VARCHAR)
	INTERVAL SECOND	Interval of seconds. The leading precision specifies the range of the integer part of the second .	INTERVAL '51.1' SECOND	Not supported (string)	Not supported (VARCHAR)

Category	Type	Description	Storage Format Example	Hive	DWS
Multimedia	IMAGE	IMAGE specifies the image type in the multimedia information. An image consists of a pixel lattice with a maximum length of 2 GB minus 1 byte. In addition to storing image data, other binary data can also be stored.	0x2A3B4058 (binary data)	Not supported	Not supported
	LONGVARBINARY	Same as IMAGE	0x2A3B4059 (binary data)	Not supported	Not supported
	TEXT	Stores the long string type. The maximum length of a string is 2 GB minus 1 byte.	0x5236 (binary data)	Not supported	Not supported
	LONGVARCHAR	Similar to TEXT	0x5236 (binary data)	Not supported	Not supported
	BLOB	Stores variable-length binary large objects with a maximum length of 2 GB minus 1 byte.	0x5236 (binary data)	Not supported	Not supported
	CLOB	Stores variable-length binary large objects with a maximum length of 2 GB minus 1 byte.	0x5236 (binary data)	Not supported	Not supported

Category	Type	Description	Storage Format Example	Hive	DWS
	BFILE	Specified the binary files stored in the operating systems. Files are stored in the operating systems instead of the databases. They can be read only.	-	Not supported	Not supported

3 Managing Clusters

3.1 Creating a CDM Cluster

Scenario

CDM provides isolated clusters to ensure secure and reliable data migration. Currently, a cluster supports only one server.

Prerequisites

You have applied for a VPC, subnet, and security group. If the CDM cluster tries to connect to another cloud service, ensure that the cluster and the cloud service are in the same VPC. Otherwise, an EIP is required.

NOTE

- If the CDM cluster and a cloud service are in the same region, VPC, subnet, and security group, they can communicate with each other through an intranet.
- If the CDM cluster and the cloud service are in the same region and VPC but in different subnets or security groups, you must configure routing rules and security group rules. For details about how to configure routing rules, see **Configuring Routes**. For details about how to configure security group rules, see **Configuring Security Group Rules**.
- If the CDM cluster and a cloud service are in different VPCs of the same region, you can create a VPC peering connection to enable them to communicate with each other. For details about how to create a VPC peering connection, see **Creating a VPC Peering Connection**.

  Note: If a VPC peering connection is created, the peer VPC subnet may overlap with the CDM management network. As a result, data sources in the peer VPC cannot be accessed. You are advised to use the Internet for cross-VPC data migration, or contact the administrator to add specific routes for the VPC peering connection in the CDM background.
- If the CDM cluster and a cloud service are located in different regions, you need to use the Internet or Direct Connect to enable them to communicate with each other. 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.
- In addition, an enterprise project may also affect the communication between the CDM cluster and other cloud services. The CDM cluster can communicate with a cloud service only if they have the same enterprise project.

Procedure

Step 1 Go to the **Buy CDM Cluster** page.

Step 2 Configure the cluster parameters. **Table 3-1** describes the required parameters.

Table 3-1 Parameter description

Parameter	Example Value	Description
Region	CN-Hong Kong	Region where the CDM cluster resides. Resources in different regions cannot communicate with each other.
AZ	AZ2	For details, see **AZs**.
Name	cdm-aff1	Custom CDM cluster name **NOTE** After a CDM cluster is created, its name cannot be changed.

Parameter	Example Value	Description
Instance Type	cdm.large	Currently, the following flavors are available: • **cdm.large**: 8 vCPUs and 16 GB of memory. The maximum and assured bandwidths are 3 Gbit/s and 0.8 Gbit/s. Up to 16 jobs can be executed concurrently. • **cdm.xlarge**: 16 vCPUs and 32 GB of memory. The maximum and assured bandwidths are 10 Gbit/s and 4 Gbit/s. Up to 32 jobs can be executed concurrently. This flavor is suitable for migrating terabytes of data that requires a bandwidth of 10GE. • **cdm.4xlarge**: 64 vCPUs and 128 GB of memory. The maximum and assured bandwidths are 40 Gbit/s and 36 Gbit/s. Up to 128 jobs can be executed concurrently. • The free ECS with 4 vCPUs and 8 GB memory provided by DataArts Studio can run only one job.
VPC	vpc1	VPC, subnet, and security group where the CDM cluster belongs to, which are used to communicate with the desired data source. They can be selected based on the migration source and destination. • If the CDM cluster and the data source to be connected belong to different VPCs or the data source is an on-premises one, the CDM cluster needs to be bound with an elastic IP address (EIP). • If the data source is a cloud service, you are advised to configure the network of the CDM cluster to be the same as that of the cloud service and the CDM cluster does not need to be bound with an EIP. • If the data source is a cloud service, and CDM and the cloud service are in the same VPC but in different subnets, configure security group rules to interconnect the CDM cluster with the cloud service. For more information, see the **Virtual Private Cloud User Guide**.
Subnet	subnet-1	
Security Group	sg-1	
Enterprise Project	default	On the management console, click **Enterprise** in the upper right corner to access the enterprise project management page to create an enterprise project. An enterprise project facilitates management of cloud resources. For more information, see the *Enterprise Management User Guide*.

Parameter	Example Value	Description
Tags	cluster_owner:cdm	Tag parameters can be configured when Advanced Configuration is set to Custom. If you want to use the same tag to identify multiple types of cloud resources, you can customize the tag key and tag value. Then, you can filter cloud resources with the same tag in the TMS tag system. **NOTE** • A cluster can have a maximum of 10 tags. • A tag key and a tag value can contain a maximum of 36 and 43 characters, respectively.
Notification	No	After the function is enabled, configure a maximum of 20 mobile numbers or email addresses. You will be notified of job failures (only table/file migration jobs) and EIP exceptions by SMS message or email.

Step 3 Check the current configuration and click **Buy Now** to go to the page for confirming the order.

NOTE

You cannot modify the flavor of an existing cluster. If you require a higher flavor, create a cluster with your desired flavor.

Step 4 Click **Submit**. The system starts to create a CDM cluster. You can view the creation progress on the **Cluster Management** page.

----End

3.2 Binding or Unbinding an EIP

Scenario

After creating a CDM cluster, you can bind an EIP to or unbind an EIP from the cluster. The EIP is billed based on the VPC service.

If CDM needs to access a local or Internet data source, or a cloud service in another VPC, bind an EIP to the CDM cluster or use a NAT gateway to enable the CDM cluster to share the EIP with ECSs to access the Internet. For details, see **Adding an SNAT Rule**.

NOTE

If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

Prerequisites

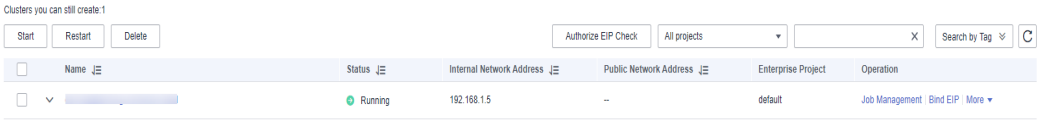
- You have created a CDM cluster.

- Your EIP quota is sufficient.

Procedure

Step 1 Log in to the CDM console and choose **Cluster Management** in the left navigation pane.

Figure 3-1 Cluster list



NOTE

The **Source** column is displayed only when you access the **DataArts Migration** page from the DataArts Studio console.

Step 2 Bind an EIP to or unbind an EIP from a cluster.

- Binding an EIP: In the **Operation** column, click **Bind EIP**. The **Bind EIP** dialog box is displayed.
- Unbinding an EIP: In the **Operation** column, choose **More** > **Unbind EIP**.

Step 3 Click **Yes**.

----End

3.3 Restarting a Cluster

Scenario

After modifying some configurations (for example, disabling user isolation), you must restart the cluster to make the modification take effect.

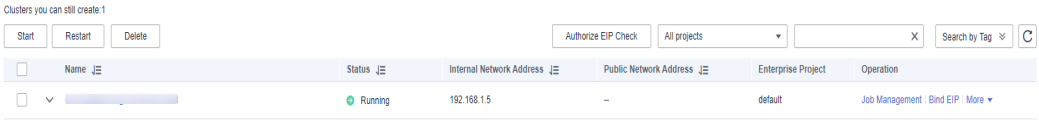
Prerequisites

You have created a CDM cluster.

Restarting a cluster

Step 1 Log in to the CDM console and choose **Cluster Management** in the left navigation pane.

Figure 3-2 Cluster list

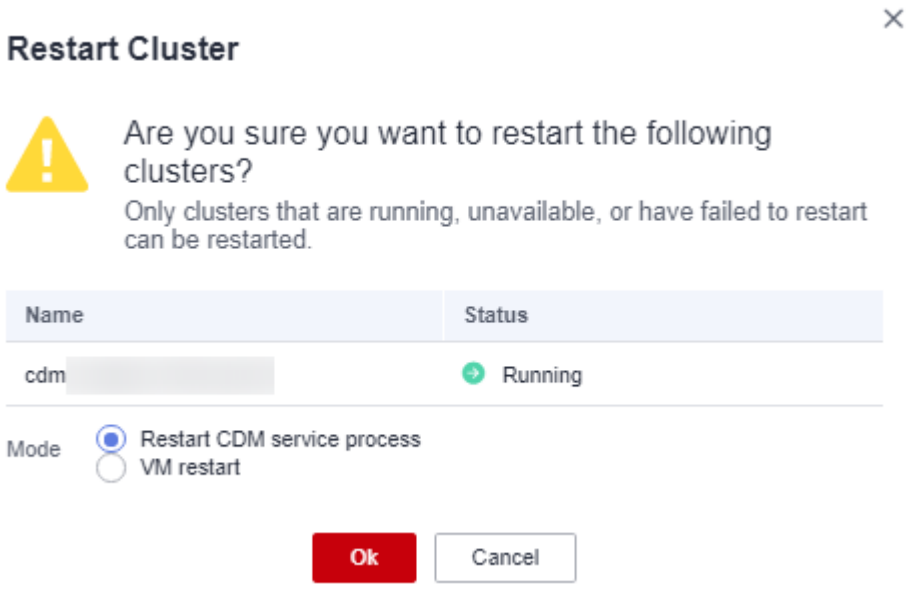


NOTE

The **Source** column is displayed only when you access the **DataArts Migration** page from the DataArts Studio console.

Step 2 Locate the row that contains the target cluster, click **More** in the **Operation** column, and select **Restart** from the drop-down list.

Figure 3-3 Restarting a cluster



Step 3 Select **Restart CDM service process** or **VM restart** and click **OK**.

- **Restart CDM service process**: Only the CDM service process is restarted. The cluster VM will not be restarted.
- **VM restart**: The service process will be interrupted and VMs in the cluster will be restarted.

----End

3.4 Deleting a Cluster

Scenario

You can delete a CDM cluster that you no longer use.

CAUTION

After a CDM cluster is deleted, the cluster and its data are destroyed and cannot be restored. Exercise caution when performing this operation.

Before deleting a cluster, note the following:

- Ensure that the cluster is not in use.
- Ensure that the links and jobs in the cluster have been backed up through the job export function described in **Managing Jobs in Batches**.
- You are not advised to delete the CDM cluster which is free of charge. If you delete it, you can only purchase clusters.
- After a CDM cluster is deleted, it will not be billed in pay-per-use mode and the package duration will not be deducted. If you have purchased a CDM discount package or a yearly/monthly CDM incremental package for the CDM cluster to delete, unsubscribe from the package by following the instructions in **Unsubscriptions**.

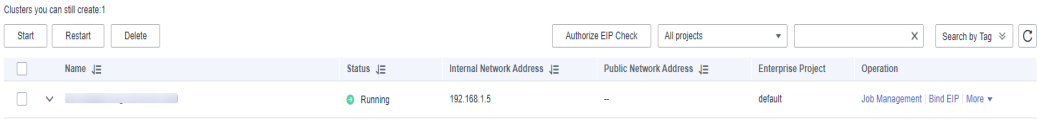
Prerequisites

You have created a CDM cluster.

Deleting a Cluster

Step 1 Log in to the CDM console and choose **Cluster Management** in the left navigation pane.

Figure 3-4 Cluster list



NOTE

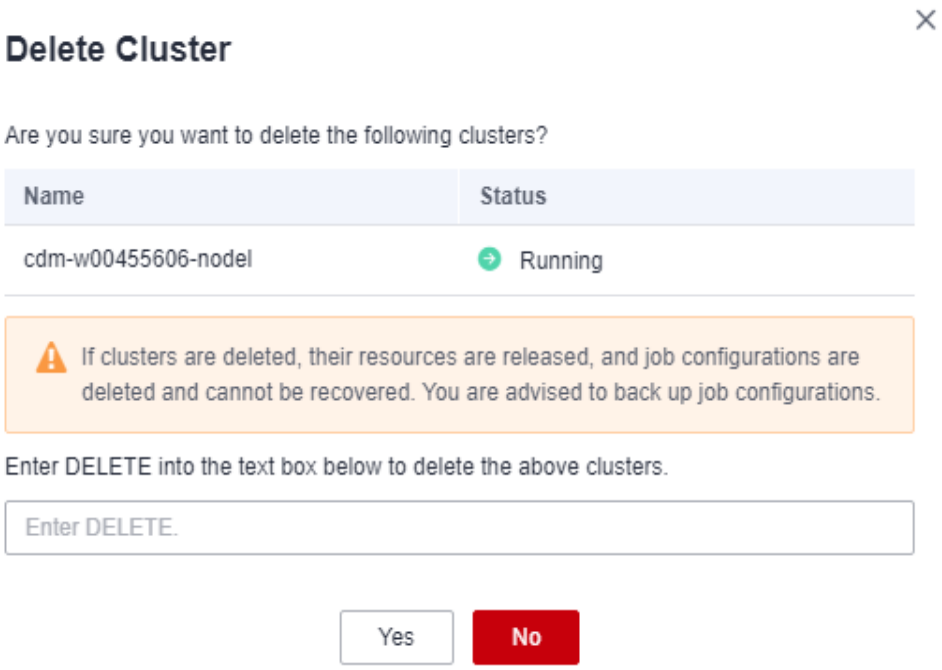
The **Source** column is displayed only when you access the **DataArts Migration** page from the DataArts Studio console.

Step 2 Delete a cluster using either of the following methods:

- Locate a cluster, click **More** in the **Operation** column, and select **Delete**.
- Select a cluster and click **Delete** above the cluster list.

Step 3 Enter **DELETE** and click **Yes**.

Figure 3-5 Deleting a cluster



----End

3.5 Downloading Cluster Logs

Scenario

This section describes how to obtain cluster logs to view the job running history and locate job failure causes.

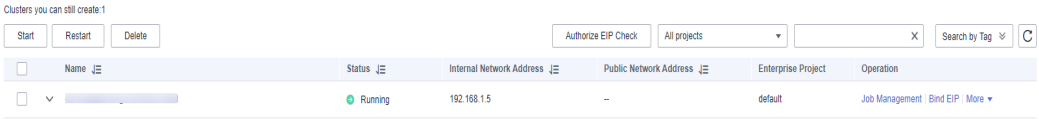
Prerequisites

You have created a CDM cluster.

Procedure

Step 1 Log in to the CDM console and choose **Cluster Management** in the left navigation pane.

Figure 3-6 Cluster list

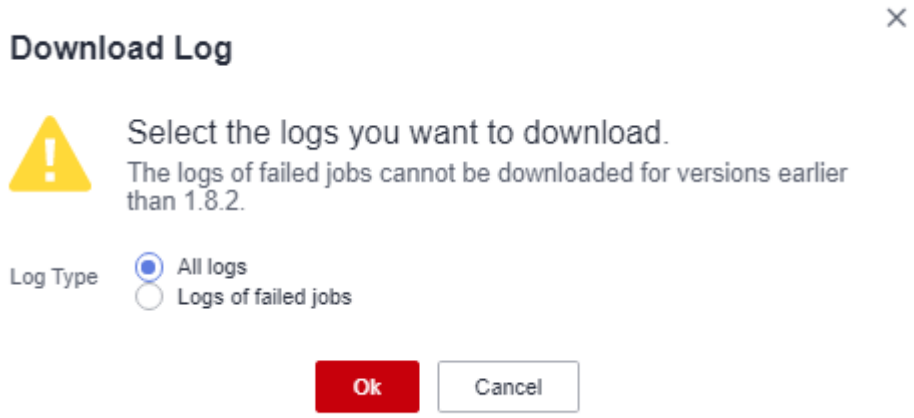


NOTE

The **Source** column is displayed only when you access the **DataArts Migration** page from the DataArts Studio console.

Step 2 Locate the row that contains a cluster, click **More** in the **Operation** column, and select **Download Log** from the drop-down list.

Figure 3-7 Download Log



Step 3 In the displayed dialog box, click **OK** to download logs to a local PC.

----End

3.6 Viewing Basic Cluster Information and Modifying Cluster Configurations

Scenario

After creating a CDM cluster, you can view its basic information and modify its configurations.

- You can view the following basic cluster information:
  - Cluster information: cluster version, creation time, project ID, instance ID, and cluster ID
  - Instance configuration: cluster flavor, CPU, and memory
  - Network configuration
- You can modify the following cluster configurations:
  - Notification: If a CDM migration job (only table/file migration) fails or the EIP is abnormal, CDM sends an SMS or email notification to the user. Notifications generated by this function will not be charged.
  - User isolation: determines whether other users can operate the migration jobs or links in the cluster.
    - If this function is enabled, migration jobs and links in the cluster are isolated. Other IAM users of the a Huawei account cannot operate the jobs and links.

- If this function is disabled, migration jobs and links in the cluster can be shared by users. All IAM users with the required permission in the a Huawei account can view and perform operations on the jobs and links in the cluster.

  After disabling **User Isolation**, restart the cluster VM for the settings to take effect.

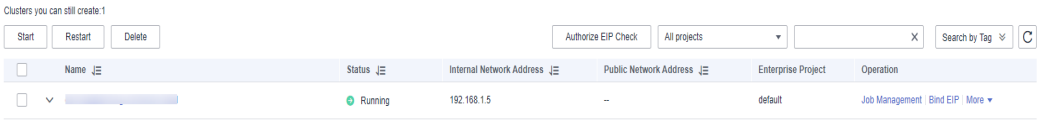
Prerequisites

You have created a CDM cluster.

Viewing Basic Cluster Information

Step 1 Log in to the CDM console and choose **Cluster Management** in the left navigation pane.

Figure 3-8 Cluster list



NOTE

The **Source** column is displayed only when you access the **DataArts Migration** page from the DataArts Studio console.

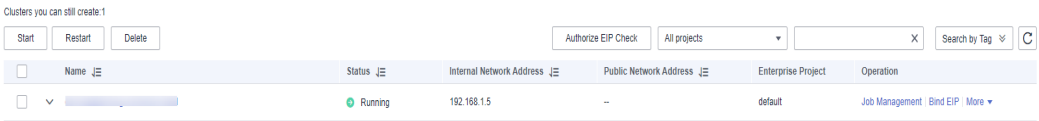
Step 2 Click the cluster name to view its basic information.

----End

Modifying Cluster Configurations

Step 1 Log in to the CDM console and choose **Cluster Management** in the left navigation pane.

Figure 3-9 Cluster list



NOTE

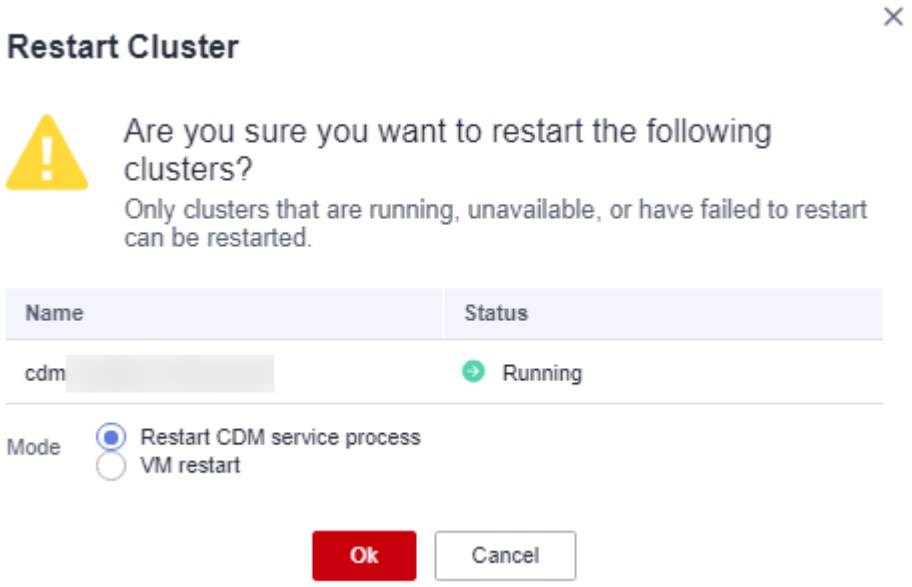
The **Source** column is displayed only when you access the **DataArts Migration** page from the DataArts Studio console.

Step 2 Click the name of a cluster and click the **Cluster Configuration** tab to modify **Notification** and **User Isolation** configuration.

Step 3 Click **Save**. The **Cluster Management** page is displayed.

Step 4 If **User Isolation** is disabled, choose **More** > **Restart** in the **Operation** column to restart the cluster VM for the settings to take effect.

Figure 3-10 Restarting a cluster



- **Restart CDM service process**: Only the CDM service process is restarted. The cluster VM will not be restarted.
- **VM restart**: The service process will be interrupted and VMs in the cluster will be restarted.

Step 5 Select **VM restart** and click **Yes**.

----End

3.7 Managing Cluster Tags

Scenario

You can add, modify, and delete tags for CDM clusters. Tags can be used to identify multiple types of cloud resources. Cloud resources with the same tag can be filtered out in the TMS tag system or on the CDM **Cluster Management** page.

NOTE

A maximum of 10 tags can be added to a CDM cluster.

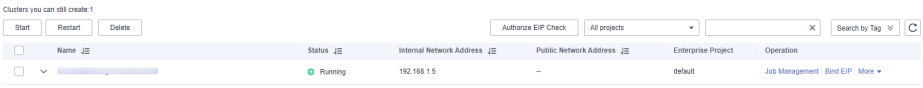
Prerequisites

You have created a CDM cluster.

Procedure

Step 1 Log in to the CDM console. In the navigation pane, choose **Cluster Management**.

Figure 3-11 Cluster list

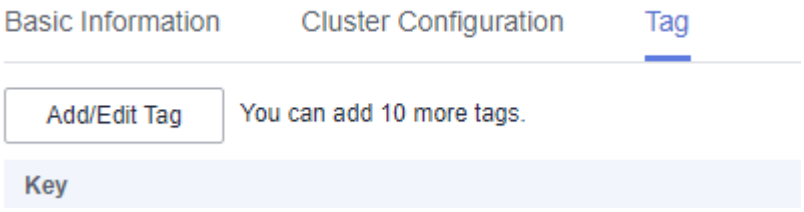


NOTE

The **Source** column is displayed only when you access the **DataArts Migration** page from the DataArts Studio console.

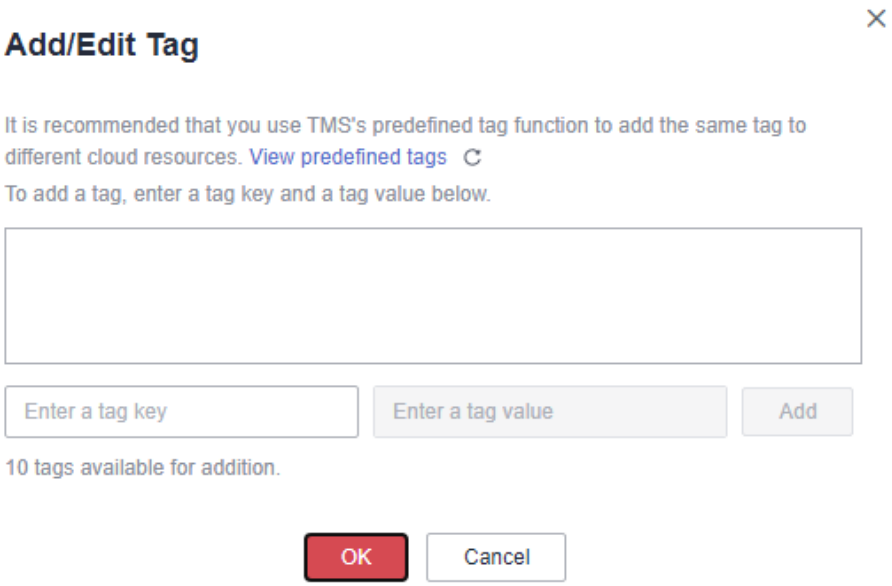
Step 2 Click a cluster name and then the **Tag** tab.

Figure 3-12 Modifying Cluster Configurations



Step 3 Click **Add/Edit Tag** and add tags to or modify tags for the CDM cluster.

Figure 3-13 Adding/Editing a tag



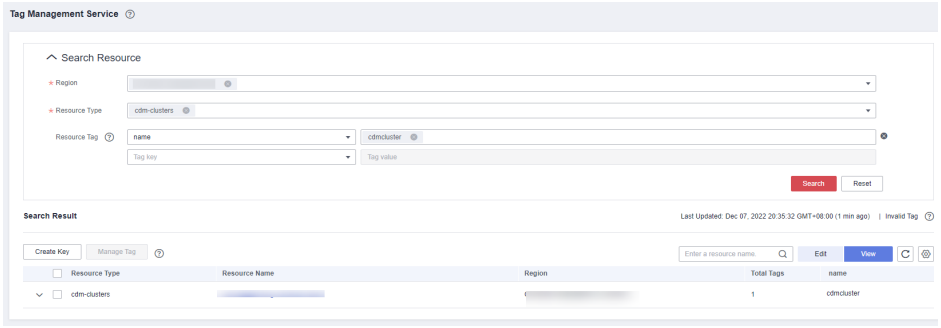
NOTE

- A cluster can have a maximum of 10 tags.
- A tag key and a tag value can contain a maximum of 36 and 43 characters, respectively.

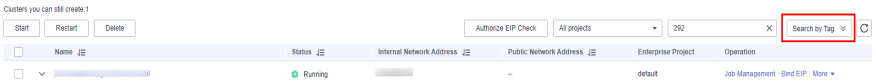
Step 4 (Optional) In the tag list, click **Delete** in the **Operation** column to delete tags.

Step 5 Use either of the following methods to filter out the resources matching specified tags:

- On the TMS console, set resource search criteria and click **Search** to obtain the clusters with the specified tags.



- On the **Cluster Management** page, click **Search by Tag**, select tags, and click **Search** to obtain the clusters with the specified tags.



----End

3.8 Viewing Metrics

3.8.1 CDM Metrics

Function

Cloud Eye monitors the running status of cloud services and usage of each metric, and creates alarm rules for monitoring metrics.

After you create a CDM cluster, Cloud Eye automatically associates with CDM monitoring metrics to help you understand the running status of the CDM cluster.

- This section describes the CDM metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions.
- For details about CDM monitoring metrics, see **Querying Metrics**.
- For details about how to set alarm rules, see **Configuring Alarm Rules**.

Prerequisites

You have obtained required Cloud Eye permissions.

Namespace

SYS.CDM

Metrics

Table 3-2 lists the CDM metrics.

Table 3-2 CDM metrics

ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
bytes_in	Bytes In	Measures the network inbound rate of the monitored object. Unit: byte/s	≥ 0 bytes/s	Cloud Data Migration	1 minute
bytes_out	Bytes Out	Measures the network outbound rate of the monitored object. Unit: byte/s	≥ 0 bytes/s	Cloud Data Migration	1 minute
cpu_usage	CPU Usage	Measures the CPU usage of the monitored object. Unit: %	0% to 100%	Cloud Data Migration	1 minute
mem_usage	Memory Usage	Measures the memory usage of the monitored object. Unit: %	0% to 100%	Cloud Data Migration	1 minute
pg_pending_job	Number of Queued Jobs	Number of jobs in the PENDING state in the CDM instance. Unit: count	>=0	Cloud Data Migration	1 minute
pending_threads	Maximum Concurrent Extractors	Number of concurrent extraction threads in the Waiting state in the CDM instance. Unit: count	>=0	Cloud Data Migration	1 minute
disk_usage	Disk Usage	Measures the disk usage of the physical server accommodating the monitored ECS, which is not accurate as that obtained on the monitored ECS. Unit: %	0.001% to 90%	Cloud Data Migration	1 minute

ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
disk_io	Disk I/O	Measures the bytes read from and written to a disk per second on the physical server accommodating the monitored ECS, which is not accurate as those obtained on the monitored ECS. Unit: Byte/s	0 GB to 10 GB	Cloud Data Migration	1 minute
tomcat_heap_usage	Heap Memory Usage	Measures the heap memory usage of the physical server accommodating the monitored ECS, which is not accurate as that obtained on the monitored ECS. Unit: %	0.001% to 90%	Cloud Data Migration	1 minute
tomcat_connect	Tomcat Concurrent Connections	Measures the number of Tomcat concurrent connections on the physical server.	0 to 2,147,483,647	Cloud Data Migration	1 minute
tomcat_thread_count	Tomat Threads	Measures the number of Tomcat threads on the physical server.	0 to 2,147,483,647	Cloud Data Migration	1 minute
pg_connect	Database Connections	Measures the number of Postgres database connections on the physical server.	0 to 2,147,483,647	Cloud Data Migration	1 minute
pg_submission_row	Rows	Measures the number of rows in the submission table of the Postgres database on the physical server.	0 to 2,147,483,647	Cloud Data Migration	1 minute
pg_failed_job_rate	Job Failure Rate	Measures the job failure rate of the sqoop process on the physical server. Unit: %	0.001% to 100%	Cloud Data Migration	1 minute

ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
inodes_usage	Inodes Usage	Measures the disk inodes usage of the physical server accommodating the monitored ECS, which is not accurate as that obtained on the monitored ECS. Unit: %	0.001% to 0.9%	Cloud Data Migration	1 minute

Dimension

Key	Value
instance_id	CDM instance

3.8.2 Configuring Alarm Rules

Scenario

Set the alarm rules to customize the monitored objects and notification policies. Then, learn CDM running status in a timely manner.

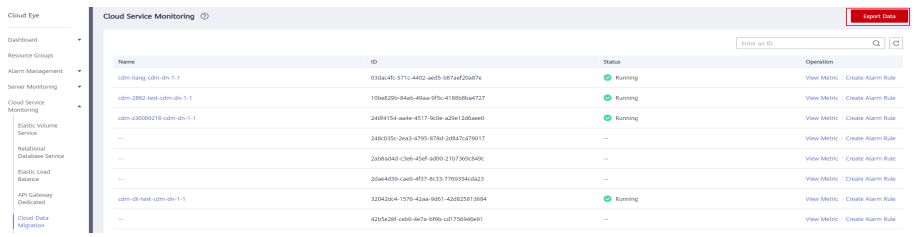
A CDM alarm rule includes the alarm rule name, monitored object, metric, threshold, monitoring interval, and whether to send a notification. This section describes how to set CDM alarm rules.

Procedure

Step 1 Access the CDM console, choose **Cluster Management**. Locate a cluster, click **More** in the **Operation** column, and select **View Metric** from the drop-down list.

Step 2 In the navigation pane, choose **Cloud Service Monitoring** > **Cloud Data Migration**. In the right pane, locate a CDM cluster and click **Create Alarm Rule** in the **Operation** column.

Figure 3-14 Monitored CDM clusters



Step 3 Set the alarm rule for the CDM cluster as prompted.

Step 4 After the setting is complete, click **Confirm**. When an alarm that meets the rule is generated, the system automatically sends a notification.

NOTE

For more information about monitoring and alarms, see the ***Cloud Eye User Guide***.

----End

3.8.3 Querying Metrics

Scenario

You can use Cloud Eye to monitor the running status of a CDM cluster. You can view the monitoring metrics on the Cloud Eye console.

Monitored data takes some time for transmission and display. The status displayed on the Cloud Eye console is the status obtained 5 to 10 minutes before. You can view the monitored data of a newly created CDM cluster 5 to 10 minutes later.

Prerequisites

- The CDM cluster is running properly.

  If a cluster fails to be restarted or is unavailable, its monitoring metrics are unavailable. You can view the monitored data only after the cluster is restarted or recovered.
- The cluster has been properly running for about 10 minutes.

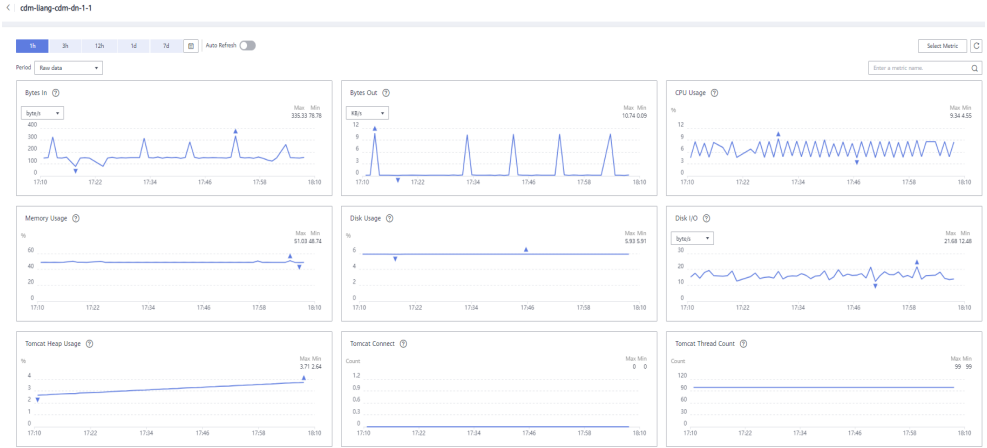
  The monitored data and graphs are available for a newly created cluster after the cluster runs for at least 10 minutes.

Procedure

Step 1 Access the CDM console, choose **Cluster Management**. Locate a cluster, click **More** in the **Operation** column, and select **View Metric** from the drop-down list.

Step 2 On the CDM monitoring page, you can view the graphs of all monitoring metrics.

Figure 3-15 Querying Metrics



Step 3 Click  in the upper right corner of the graphs to zoom in the graphs.

Step 4 You can select a time period in the upper left corner to view metric changes in this time period.

----End

4 Managing Links

4.1 Creating a Link

Scenario

Before creating a data migration job, create a link to enable the CDM cluster to read data from and write data to a data source. A migration job requires a source link and a destination link. For details on the data sources that can be exported (source links) and imported (destination links) in different migration modes (table/file migration), see **Supported Data Sources**.

The link configurations depend on the data source. This section describes how to create these links.

Constraints

If changes occur in the connected data source (for example, the MRS cluster capacity is expanded), you need to edit and save the connection.

Prerequisites

- A CDM cluster is available.
- The CDM cluster can communicate with the destination data source.
  - 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 destination data source is a cloud service (such as DWS, MRS, and ECS), the following requirements must be met for network interconnection:
    - 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.

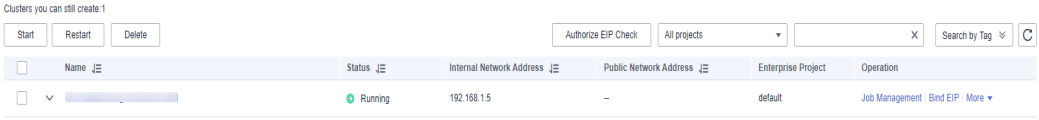
- If the CDM cluster and the cloud service are in the same region, VPC, subnet, and security group, they can communicate with each other by default. If they are in the same VPC but in different subnets or security groups, you must configure routing rules and security group rules. For details about how to configure routing rules, see **configuring routes**. For details about how to configure security group rules, see **configuring security group rules**.
- 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.

- You have obtained the URL and the account for accessing the data source. The account is granted with the read and write permissions for the data source.
- When using the Agent, you need to use the main account to grant the CDM operation permission to the sub-account.

Creating Links

Step 1 Log in to the CDM console and choose **Cluster Management** in the left navigation pane.

Figure 4-1 Cluster list



NOTE

The **Source** column is displayed only when you access the **DataArts Migration** page from the DataArts Studio console.

Step 2 On the CDM console, choose **Cluster Management** in the left navigation pane. Locate the row that contains the target cluster and click **Job Management** in the **Operation** column. On the displayed **Links** page, click **Create Link**. On the displayed page shown in **Figure 4-2**, select a connector.

The connectors are classified based on the type of the data source to be connected. All supported data types are displayed.

Figure 4-2 Selecting a connector type

Step 3 Select a data source and click **Next**. The following describes how to create a MySQL link.

The link parameters of different data sources vary. **Table 4-1** describes the link parameters.

Table 4-1 Link parameters

Connector	Description
• RDS for PostgreSQL • RDS for SQL Server • PostgreSQL • Microsoft SQL Server	Because the JDBC drivers used by these relational databases are the same, the parameters to be configured are also the same and are described in **Link to PostgreSQL/ SQLServer**.
Data Warehouse Service	For details about the parameters, see **Link to DWS**.
SAP HANA	For details about the parameters, see **Link to SAP HANA**.
Dameng database	For details about the parameters, see **Link to a Dameng Database**.
MySQL	For details about the parameters, see **Link to an RDS for MySQL/MySQL Database**.
Oracle	For details about the parameters, see **Link to an Oracle Database**.
Database Sharding	For details about the parameters, see **Link to a Database Shard**.

Connector	Description
Object Storage Service (OBS)	For details about the parameters, see **Link to OBS**.
• MRS HDFS • FusionInsight HDFS • Apache HDFS	If the data source is HDFS of MRS, Apache Hadoop, or FusionInsight HD, see **Link to HDFS**.
• MRS HBase • FusionInsight HBase • Apache HBase	If the data source is HBase of MRS, Apache Hadoop, or FusionInsight HD, see **Link to HBase**.
• MRS Hive • FusionInsight Hive • Apache Hive	If the data source is Hive on MRS, Apache Hadoop, or FusionInsight HD, see **Link to Hive**.
CloudTable Service	If the data source is CloudTable, see **Link to CloudTable**.
• FTP • SFTP	If the data source is an FTP or SFTP server, see **Link to an FTP or SFTP Server**.
HTTP	These connectors are used to read files with an HTTP/HTTPS URL, such as reading public files on the third-party object storage system and web disks. When creating an HTTP link, you only need to configure the link name. The URL is configured during job creation.
MongoDB	If the data source is a local MongoDB, see **Link to MongoDB**.
Document Database Service (DDS)	If the data source is DDS, see **Link to DDS**.
• Redis • Distributed Cache Service	If the data source is Redis or DCS, see **Link to Redis**.
• MRS Kafka • Apache Kafka	If the data source is MRS Kafka or Apache Kafka, see **Link to Kafka**.
Data Ingestion Service	If the data source is DIS, see **Link to DIS**.
Cloud Search Service (CSS) Elasticsearch	If the data source is CSS or Elasticsearch, see **Link to CSS**.
Data Lake Insight	If the data source is DLI, see **Link to DLI**.
DMS Kafka	If the data source is DMS Kafka, see **Link to DMS Kafka**.

Connector	Description
Cassandra	If the data source is Cassandra, see **Link to Cassandra**.
MRS Hudi	For details about the parameters, see **Link to MRS Hudi**.
MRS ClickHouse	For details about the parameters, see **Link to MRS ClickHouse**.
Shentong database	For details about the parameters, see **Link to a ShenTong Database**.

NOTE

Currently, the following data sources are in the OBT phase: FusionInsight HDFS, FusionInsight HBase, FusionInsight Hive, SAP HANA, Document Database Service, CloudTable Service, Cassandra, DMS Kafka, Cloud Search Service, Sharding Database, and ShenTong Database.

Step 4 After configuring the parameters of the link, click **Test** to check whether the link is available. Alternatively, click **Save**, and the system checks automatically.

If the network is poor or the data source is too large, the link test may take 30 to 60 seconds.

----End

Managing Links

CDM allows you to perform the following operations on created links:

- Deleting links: You can delete links that are not used by any job.
- Editing a link: You can modify link parameters but cannot reselect the connector. To modify a link, you need to re-enter the password needed to access the data source.
- Testing connectivity: You can test connectivity of a link that has been saved.
- Viewing the JSON file of a link: You can view parameters of a link in a JSON file.
- Editing the JSON file of a link: Modify parameters of a link in a JSON file.
- Viewing the backend link: You can view the backend link corresponding to a link. For example, you can query details about the backend link if it is enabled.

Before managing a link, ensure that the link is not used by any job to avoid affecting job execution. The procedure for managing connections is as follows:

Step 1 Log in to the management console and choose **Service List** > **Cloud Data Migration**. On the CDM console, choose **Cluster Management** in the left navigation pane. Locate the row that contains the target cluster and click **Job Management** in the **Operation** column. On the displayed page, click the **Links** tab.

Step 2 On the **Links** page, locate the link to be modified.

- Deleting a link: Click **Delete** in the **Operation** column to delete a link. Alternatively, select the links that are not used by any job and click **Delete Link** above the list to delete them.
- Editing the link: Click the link name or click **Edit** in the **Operation** column to access the page for modifying the link. When modifying the link, you need to enter the password for logging in to the data source again.
- Testing connectivity of the link: Click **Test Connectivity** in the **Operation** column.
- Viewing the JSON file of the link: In the **Operation** column, choose **More** > **View Link JSON** to view link parameters in JSON format.
- Editing the JSON file of the link: In the **Operation** column, choose **More** > **Edit Link JSON** to modify link parameters in JSON format.
- Viewing the backend link: Locate the row that contains a link and click **More** in the **Operation** column and select **View Backend Link** to view the backend link corresponding to the link.

----End

4.2 Managing Drivers

The Java Database Connectivity (JDBC) provides programmatic access to relational databases. Applications can execute SQL statements and retrieve data using the JDBC API.

Before connecting CDM to a relational database, you need to upload the JDK 8 .jar driver of the relational database.

Prerequisites

- A cluster has been created.
- You have downloaded one of the drivers listed in **Table 4-2**.
- (Optional) An SFTP link has been created by referring to **Link to an FTP or SFTP Server** and the corresponding driver has been uploaded to the offline file server.

How Do I Obtain a Driver?

Select a driver version that adapts to the database type. Note that the version of the uploaded driver does not need to match the version of the database to be connected. Obtain the JDK8 .jar driver of the recommended version by referring to **Table 4-2**.

Table 4-2 Drivers

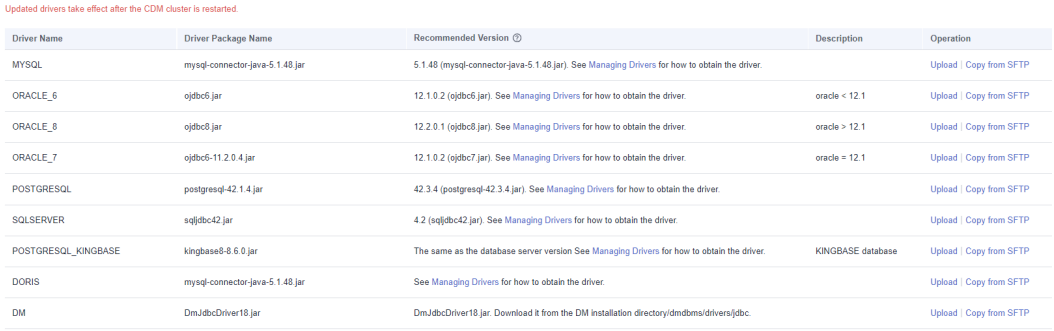
Relational Database Type	Driver Name	How to Obtain	Recommended Version
• RDS for MySQL • MySQL	MySQL	**https://downloads.mysql.com/archives/c-j/**	mysql-connector-java-5.1.48.jar
Oracle	ORACLE_6 ORACLE_7 ORACLE_8	Driver packages: **https://www.oracle.com/database/technologies/appdev/jdbc-downloads.html** Driver packages of historical versions: **https://repo1.maven.org/maven2/com/oracle/database/jdbc/**	ojdbc8.jar for version 12.2.0.1 **NOTE** New versions (for example, Oracle Database 21c (21.3) drivers) are not supported. If they are used, the schema name cannot be obtained during job creation.
• RDS for PostgreSQL • PostgreSQL	POSTGRESQL	**https://mvnrepository.com/artifact/org.postgresql/postgresql**	postgresql-42.3.4.jar for version 42.3.4
KingBase	POSTGRESQL	**https://mvnrepository.com/artifact/org.postgresql/postgresql**	postgresql-42.2.9.jar for PostgreSQL 42.2.9
GaussDB	POSTGRESQL	GaussDB JDBC driver: Search for "JDBC Package, Driver Class, and Environment Class" in **GaussDB Documentation**, select the document corresponding to the instance version, and obtain **gsjdbc4.jar** by referring to the document.	Obtain **gsjdbc4.jar** from the release package of the corresponding version.

Relational Database Type	Driver Name	How to Obtain	Recommended Version
• RDS for SQL Server • Microsoft SQL Server	SQLServer	https://docs.microsoft.com/en-us/sql/connect/jdbc/release-notes-for-the-jdbc-driver?view=sql-server-ver15#previous-releases	sqljdbc42.jar

Procedure

Step 1 Access the CDM console, choose **Cluster Management** in the navigation pane, locate the target cluster, and choose **Job Management** > **Link Management** > **Driver Management**. On the **Driver Management** page, upload a driver.

Figure 4-3 Uploading a driver



Step 2 Click **Upload** in the **Operation** column and select a local driver.

Alternatively, click **Copy from SFTP** in the **Operation** column and configure the **SFTP Link** name and **Driver File Path**.

Step 3 (Optional) If you have uploaded an updated version of a driver, you must restart the CDM cluster for the new driver to take effect.

----End

4.3 Managing Agents

If your data is stored in HDFS or a relational database, you can deploy an agent on the source network. CDM pulls data from your internal data sources through an agent but cannot write data into the databases.

Figure 4-4 Scenario

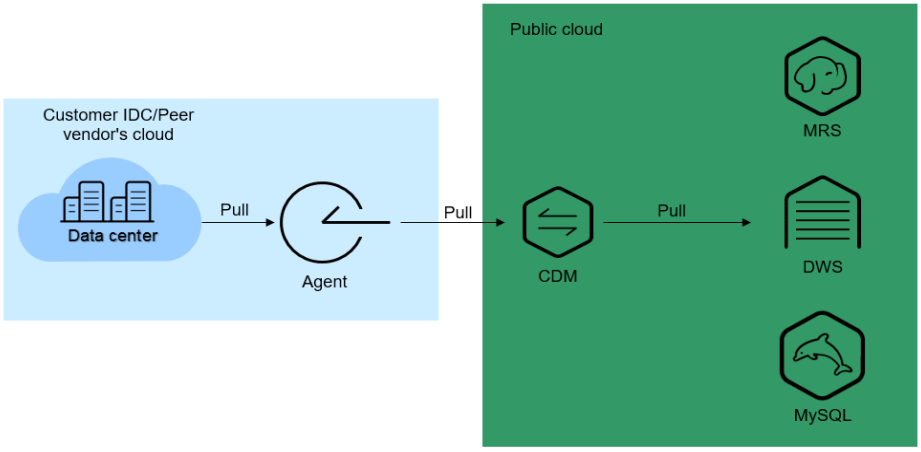
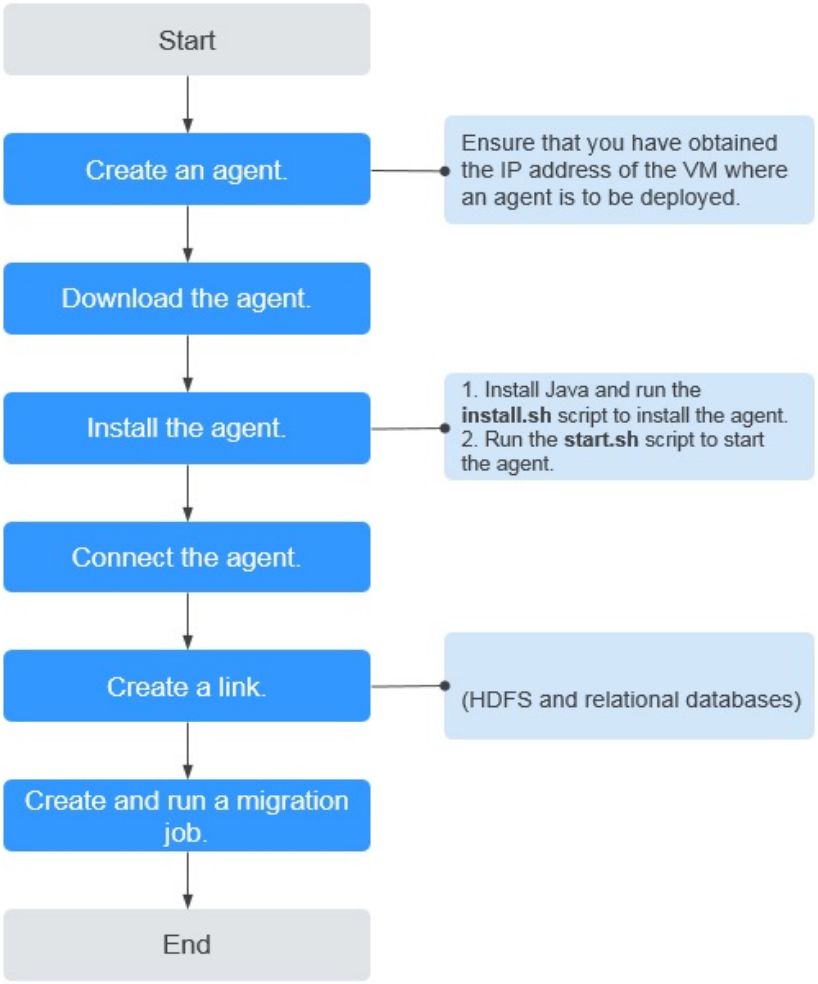


Figure 4-5 shows the process of using an agent.

Figure 4-5 Process



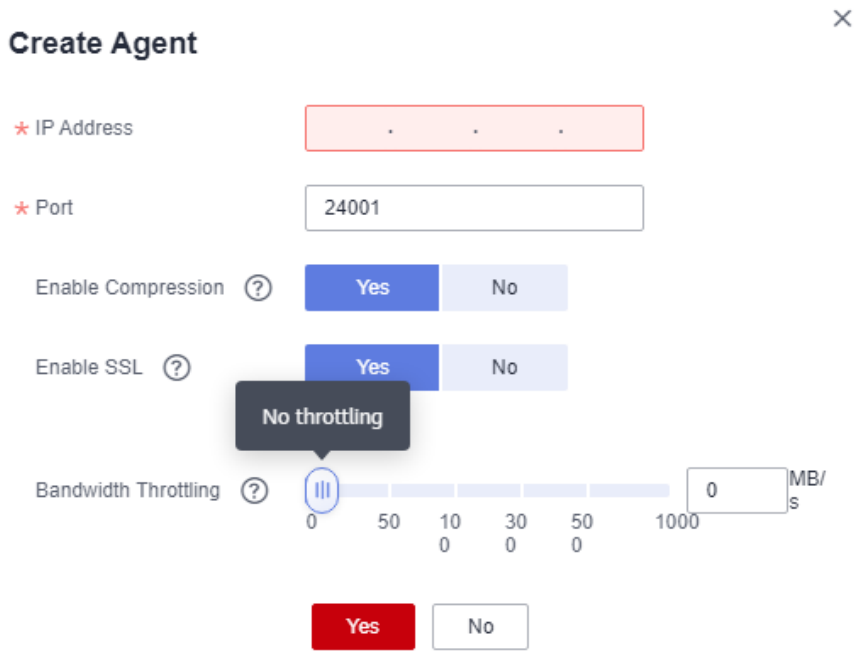
Prerequisites

- A CDM cluster is available.
- A Linux server (for example, a Linux ECS) is available. The Linux server has no special requirements for vCPUs, memory, and disks, but must meet the following requirements:
  - Java 8 (64-bit) has been installed and Java environment variables have been configured.
  - User **Ruby** must be granted the write permission of the **/tmp** directory. If there is no user **Ruby**, create one.

Creating an Agent

Step 1 Access the CDM console and choose **Cluster Management** in the left navigation pane. Locate the target cluster, choose **Job Management** > **Agent Management** > **Create Agent**, and configure agent parameters.

Figure 4-6 Creating an agent



- **IP Address**: Set this parameter to the IP address of the server where the agent is deployed on the source network.
- **Port**: custom port of the agent Recommended value range: 1024–65535.
- **Enable Compression**: whether to compress data using the gzip algorithm.
  - Enable this function for text data (data based on character encoding, such as MySQL INT data) because such data can be well compressed by the gzip algorithm. (For details about text data, see the related database documentation.)
  - Disable this function for binary data (data based on value encoding, such as MySQL BINARY data) because such data has been compressed, and compressing it again will increase the workload to decompress data and

undermine the performance of the client. (For details about text data, see the related database documentation.)

- **Enable SSL**: whether to enable two-way SSL authentication Enable this function if security is of high priority.
- **Bandwidth Throttling**: set the maximum downstream rate of the agent. By default, there is no throttling.

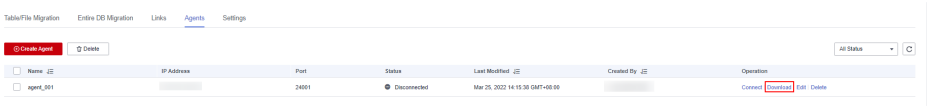
Step 2 Click **OK**. On the **Agent Management** page, view the created agent.

----End

Installing and Starting an Agent

Step 1 On the **Agent Management** page, locate the created agent and click **Download** in the **Operation** column.

Figure 4-7 Downloading an agent



Step 2 Upload the downloaded agent package to the target Linux server.

NOTE

The Linux server has no special requirements for vCPUs, memory, and disks, but must meet the following requirements:

- Java 8 (64-bit) has been installed and Java environment variables have been configured.
- User **Ruby** must be granted the write permission of the **/tmp** directory. If there is no user **Ruby**, create one.

Step 3 Decompress the package and run the following command to install the agent:

sh sbin/install.sh

Step 4 If you want to use the agent to connect to a relational database, you need to upload the corresponding drivers (see **Managing Drivers**) to the **/server/jdbc** directory in the agent installation directory and modify the version number of the corresponding database driver in the **properties** file in the same directory.

Step 5 Run the following command as user **root** to change the owner and group of the driver uploaded to the **/server/jdbc** directory to **Ruby**:

chown Ruby.Ruby * -R

Step 6 After the installation is complete, run the following commands to start the agent:

su Ruby

sh sbin/start.sh

Step 7 Run the following command to check whether the agent is started:

ps -ef | grep cdm

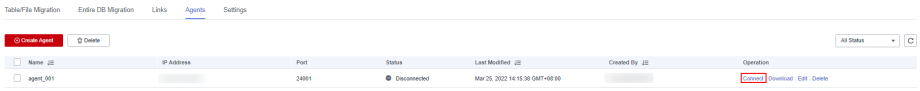
If the command output contains the running agent process, the agent process has been started.

----End

Connecting to an Agent

Step 1 On the **Agent Management** page, locate the created agent and click **Connect** in the **Operation** column.

Figure 4-8 Connecting to an agent



Step 2 After the agent is successfully connected, you can select it when creating a connection.

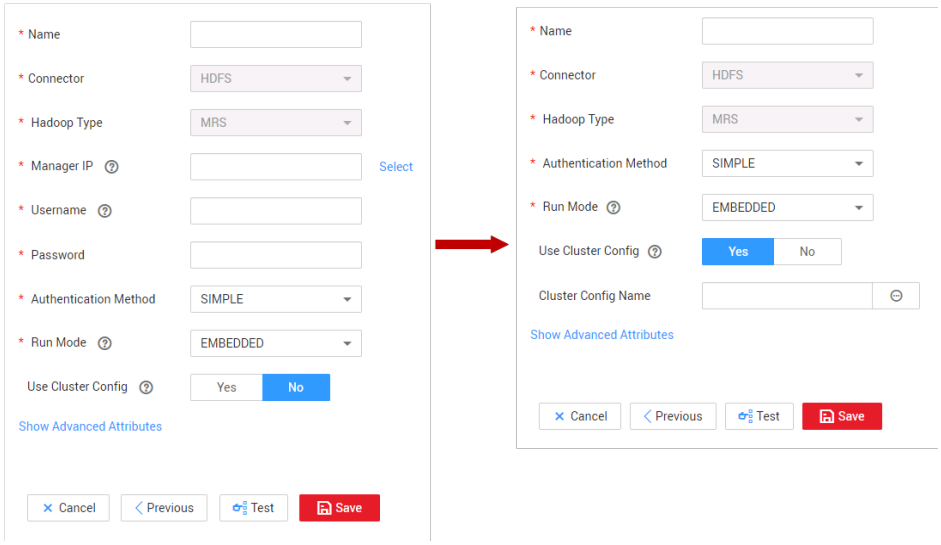
----End

4.4 Managing Cluster Configurations

On the **Cluster Configurations** page, you can create, edit, or delete Hadoop cluster configurations.

When creating a Hadoop link, the Hadoop cluster configurations can simplify the link creation. See **Figure 4-9** for details.

Figure 4-9 Comparison before and after using the cluster configurations



CDM supports the following types of Hadoop links:

- MRS clusters: MRS HDFS, MRS HBase, and MRS Hive
- FusionInsight clusters: FusionInsight HDFS, FusionInsight HBase, and FusionInsight Hive

- Apache clusters: Apache HDFS, Apache HBase, and Apache Hive

Scenario

Before creating a Hadoop link, you are advised to create cluster configurations to simplify the link parameter configurations.

Prerequisites

- A cluster has been created.
- You have obtained the Hadoop cluster configuration file and keytab file. See **Table 1** for details.

Obtaining the Cluster Configuration File and Keytab File

The methods for obtaining the Hadoop cluster configuration file and keytab file vary depending on the Hadoop cluster type. For details, see **Table 1**.

Table 4-3 Obtaining the cluster configuration file and keytab file

Hadoop Link	Obtaining the Cluster Configuration File	Obtaining the Keytab File
MRS cluster • MRS HDFS • MRS HBase • MRS Hive • MRS Hudi • MRS ClickHouse	For clusters of MRS 3.x: 1. Log in to FusionInsight Manager. 2. Choose **Cluster** > *Name of the desired cluster* > **Dashboard** > **More** > **Download Client**. 3. In the dialog box that is displayed, select **Configuration Files Only**. The platform type must be the same as that on the server. Retain the default values of other parameters and click **OK** to download the configuration file to the local host. 4. Obtain the downloaded TAR package, which is the FusionInsight cluster configuration file. For clusters of MRS 2.x or earlier: 1. Log in to the MRS console. 2. Choose **Clusters** > **Active Clusters** and click a cluster name to go to the cluster details page. Click the **Components** tab. 3. Click **Download Client**. Set **Client Type** to **Only configuration files**, set **Download To** to **Server** or **Remote host**, customize the client path, and click **OK** to generate the client configuration file. 4. Save the generated configuration file to a local path. See MRS documentation for details.	For clusters of MRS 3.x: 1. Log in to FusionInsight Manager. 2. Choose **System** > **Permission** > **User**, locate the row that contains the target user, and choose **More** > **Download Authentication Credential** to download the authentication credential file. 3. Obtain the downloaded TAR package, which is the keytab file of the FusionInsight cluster. For clusters of MRS 2.x or earlier: 1. Log in to MRS Manager and click **System**. In the **Permission** area, click **Manage User**. 2. In the row of the user for whom you want to export the keytab file, choose **More** > **Download authentication credential** to download the authentication file. After the file is automatically generated, save it to a specified path and keep it properly. See MRS documentation for details.

Hadoop Link	Obtaining the Cluster Configuration File	Obtaining the Keytab File
FusionInsight clusters: • FusionInsight HDFS • FusionInsight HBase • FusionInsight Hive	1. Log in to FusionInsight Manager. 2. Choose **Cluster** > *Name of the desired cluster* > **Dashboard** > **More** > **Download Client**. 3. In the dialog box that is displayed, select **Configuration Files Only**. The platform type must be the same as that on the server. Retain the default values of other parameters and click **OK** to download the configuration file to the local host. 4. Obtain the downloaded TAR package, which is the FusionInsight cluster configuration file. See the FusionInsight documentation for details.	1. Log in to FusionInsight Manager. 2. Choose **System** > **Permission** > **User**, locate the row that contains the target user, and choose **More** > **Download Authentication Credential** to download the authentication credential file. 3. Obtain the downloaded TAR package, which is the keytab file of the FusionInsight cluster. See the FusionInsight documentation for details.

Hadoop Link	Obtaining the Cluster Configuration File	Obtaining the Keytab File
Apache clusters: • Apache HDFS • Apache HBase • Apache Hive	In the Apache cluster scenario, only the required configuration files and packaging rules are described. For details about how to obtain each configuration file, see the corresponding documentation. • HDFS needs to compress the following files into a .zip package without the directory format: – hosts – core-site.xml – hdfs-site.xml – yarn-site.xml – mapred-site.xml – krb5.conf (optional, for clusters in security mode) • HBase needs to compress the following files into a .zip package without the directory format: – hosts – core-site.xml – hdfs-site.xml – yarn-site.xml – mapred-site.xml – hbase-site.xml – krb5.conf (optional, for clusters in security mode) • Hive needs to compress the following files into a .zip package without the directory format: – hosts – core-site.xml – hdfs-site.xml – yarn-site.xml	In the Apache cluster scenario, only the principles for packaging authentication credential files are required. For details about how to obtain the authentication credential files, see the corresponding documentation. 1. Rename the user's authentication credential file as **user.keytab**. 2. Compress the **user.keytab** file into a .zip package without the directory format: **user.keytab.zip**.

Hadoop Link	Obtaining the Cluster Configuration File	Obtaining the Keytab File
	– mapred-site.xml – hive-site.xml – hivemetastore-site.xml – krb5.conf (optional, for clusters in security mode)	

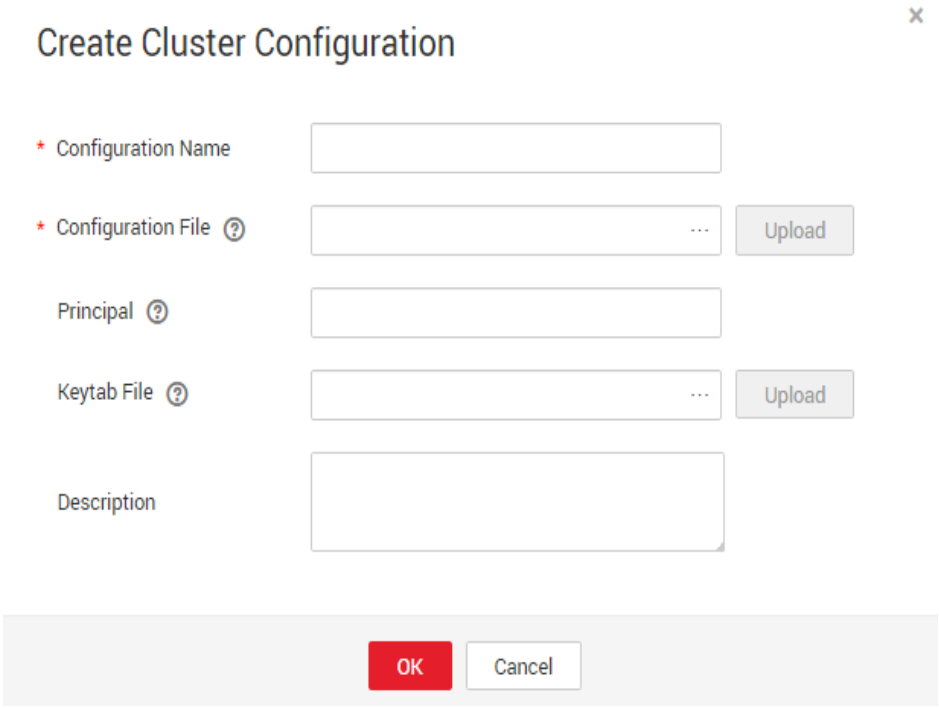
NOTE

- A cluster configuration file contains the configuration parameters of the cluster. If the cluster configuration parameters are modified, you need to obtain the configuration file again.
- The keytab file is the authentication credential file. Before obtaining the keytab file, you need to change the password of this user at least once in the cluster. Otherwise, the downloaded keytab file may be unavailable. After a user password is changed, the exported keytab file becomes invalid, and you need to export a keytab file again.
- The keytab file is used only in a cluster in security mode. In other cases, you do not need to prepare the keytab file.

Procedure

1. On the CDM console, choose **Cluster Management** in the left navigation pane. Locate the row that contains a cluster and choose **Job Management** > **Links** > **Cluster Configurations**.
2. On the **Cluster Configurations** page, click **Create Cluster Configuration** and set the parameters as prompt.

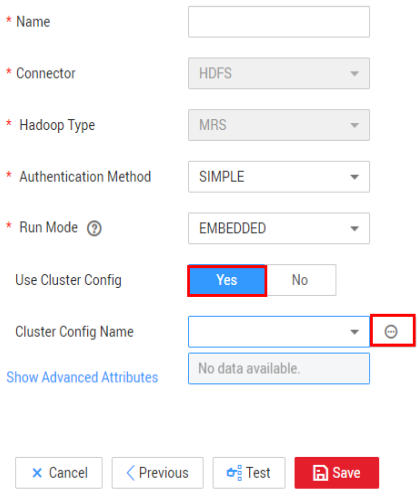
Figure 4-10 Creating cluster configurations



- **Configuration Name**: Enter a cluster configuration name that is easy to remember and distinguish based on the type of the data source to be connected.
- **Configuration File**: Click **Select File** to select a local cluster configuration file, and then click **Upload** on the right to upload the file.
- **Principal**: This parameter is required only for clusters in security mode. Principal is the username in Kerberos security mode and must be the same as that in the keytab file.
- **Keytab File**: Upload the keytab file only for clusters in security mode. Click **Select File** to select a local keytab file, and then click **Upload** on the right to upload the file.
- **Description**: Add a description to identify and distinguish the cluster configuration.

3. Click **OK**. When creating a Hadoop link, set **Authentication Method** as required, **Use Cluster Config** to **Yes**, and then select the corresponding cluster configuration name to quickly create a Hadoop link.

Figure 4-11 Use Cluster Config



4.5 Link to OBS

When connecting CDM to the destination OBS bucket, you need to add the read and write permissions to the destination OBS bucket, and file authentication is not required.

NOTE

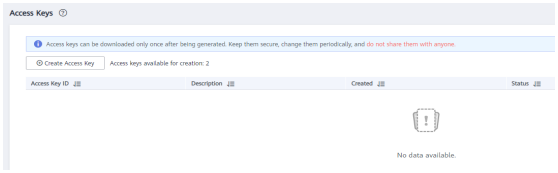
If the CDM cluster and OBS bucket are not in the same region, the CDM cluster cannot access the OBS bucket.

When connecting CDM to OBS, configure the parameters as described in **Table 4-4**.

Table 4-4 Parameter description

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	obs_link

Parameter	Description	Example Value
OBS Endpoint	An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. You can obtain the OBS bucket endpoint by either of the following means: To obtain the endpoint of an OBS bucket, go to the OBS console and click the bucket name to go to its details page. **NOTE** If the CDM cluster and OBS bucket are not in the same region, the CDM cluster cannot access the OBS bucket.	-
Port	Data transmission port. The HTTPS port number is 443 and the HTTP port number is 80.	443
OBS Bucket Type	Select a value from the drop-down list, generally, **Object Storage**.	Object Storage

Parameter	Description	Example Value
AK	AK and SK are used to log in to the OBS server. You need to create an access key for the current account and obtain an AK/SK pair. To obtain an access key, perform the following steps: 1. Log in to the management console, move the cursor to the username in the upper right corner, and select **My Credentials** from the drop-down list. 2. On the **My Credentials** page, choose **Access Keys**, and click **Create Access Key**. See **Figure 4-12**. **Figure 4-12** Clicking Create Access Key  3. Click **OK** and save the access key file as prompted. The access key file will be saved to your browser's configured download location. Open the **credentials.csv** file to view **Access Key Id** and **Secret Access Key**. **NOTE** • Only two access keys can be added for each user. • To ensure access key security, the access key is automatically downloaded only when it is generated for the first time and cannot be obtained from the management console later. Keep them properly.	-
SK		-

4.6 Link to PostgreSQL/SQLServer

Table 4-5 lists the parameters for creating a link to PostgreSQL/SQLServer. KingBase and GaussDB can be connected through the PostgreSQL connector. The source and destination data sources supported by migration jobs are the same as those for PostgreSQL..

Table 4-5 PostgreSQL/SQLServer link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	sql_link
Database Server	IP address or domain name of the database to connect Click **Select** next to the text box to obtain the list of instances.	192.168.0.1
Port	Port of the database to connect	The port number varies depending on the database. Examples: Default port of SQL Server: **1433** Default port of PostgreSQL: **5432**
Database Name	Name of the database to connect	dbname
Username	Username used for accessing the database This account must have the permissions required to read and write data tables and metadata.	cdm
Password	Password of the user	-
Use Agent	Whether to extract data from the data source through an agent	Yes
Agent	Click **Select** and select the agent created in **Managing Agents**.	-
Driver Version	Different types of relational databases adapt to different drivers. For details, see **How Do I Obtain a Driver?**	-
Fetch Size	(Optional) Displayed when you click **Show Advanced Attributes**. Number of rows obtained by each request. Set this parameter based on the data source and the job's data size. If the value is either too large or too small, the job may run for a long time.	1000

Parameter	Description	Example Value
Link Attributes	(Optional) Click **Add** to add the JDBC connector attributes of multiple specified data sources. For details, see the JDBC connector document of the corresponding database. The following are some examples: • **connectTimeout=60** and **socketTimeout=300**: When a large amount of data needs to be migrated or the entire table is retrieved using query statements, the migration fails due to connection timeout. In this case, you can customize the connection timeout interval (s) and socket timeout interval (s) to prevent failures caused by timeout. • **useCursorFetch=false**: By default, **useCursorFetch** is enabled, indicating that the JDBC connector communicates with relational databases using a binary protocol. Some third-party systems may have compatibility issues, causing migration time conversion errors. In this case, you can disable this function. • **trustServerCertificate=true**: A PKIX error may be reported during the creation of a secure connection. You are advised to set this parameter to **true**.	sslmode=require
Reference Sign	(Optional) Delimiter between the names of the referenced tables or columns. For details, see the product documentation of the corresponding database.	"

4.7 Link to DWS

Table 4-6 describes the DWS link parameters.

Table 4-6 DWS link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	dws_link

Parameter	Description	Example Value
Database Server	IP address or domain name of the database to connect Click **Select** next to the text box to obtain the list of instances.	192.168.0.1
Port	Port of the database to connect	The port number varies depending on the database.
Database Name	Name of the database to connect	dbname
Username	Username used for accessing the database. This user must have the permissions to read and write data tables and metadata.	cdm
Password	Password of the user	-
Use Agent	Whether to extract data from the data source through an agent	Yes
Agent	Click **Select** and select the agent created in **Managing Agents**.	-
Reference Sign	(Optional) Delimiter between the names of the referenced tables or columns. For details, see the product documentation of the corresponding database.	"
Fetch Size	(Optional) Displayed when you click **Show Advanced Attributes**. Number of rows obtained by each request. Set this parameter based on the data source and the job's data size. If the value is either too large or too small, the job may run for a long time.	1000
SSL Encryption	(Optional) If you set this parameter to **Yes**, CDM can connect to the database (on-premises databases excluded) in SSL encryption mode.	Yes **NOTE** To enable SSL encryption, you must ensure that it is enabled for GaussDB(DWS).

Parameter	Description	Example Value
Link Attributes	(Optional) Click **Add** to add the JDBC connector attributes of multiple specified data sources. For details, see the JDBC connector document of the corresponding database. The following are some examples: • **connectTimeout=60** and **socketTimeout=300**: When a large amount of data needs to be migrated or the entire table is retrieved using query statements, the migration fails due to connection timeout. In this case, you can customize the connection timeout interval (s) and socket timeout interval (s) to prevent failures caused by timeout. • **useCursorFetch=false**: By default, **useCursorFetch** is enabled, indicating that the JDBC connector communicates with relational databases using a binary protocol. Some third-party systems may have compatibility issues, causing migration time conversion errors. In this case, you can disable this function. Open-source MySQL databases support the **useCursorFetch** parameter, and you do not need to set this parameter.	sslmode=require **NOTE** If SSL encryption is enabled but **sslmode** is not set, the link may fail.

4.8 Link to an RDS for MySQL/MySQL Database

Table 4-7 lists the parameters for a link to a MySQL database.

Table 4-7 MySQL database link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	mysql_link
Database Server	IP address or domain name of the database to connect Click **Select** next to the text box and select a MySQL DB instance in the displayed dialog box.	192.168.0.1
Port	Port of the database to connect	3306
Database Name	Name of the database to connect	dbname

Parameter	Description	Example Value
Username	Username used for accessing the database This account must have the permissions required to read and write data tables and metadata.	cdm
Password	Password of the user	-
Use Local API	(Optional) Whether to use the local API of the database for acceleration. When you create a MySQL link, CDM automatically enables the **local_infile** system variable of the MySQL database to enable the LOAD DATA function, which accelerates data import to the MySQL database. If this parameter is enabled, the date type that does not meet the format requirements will be stored as 0000-00-00. For details, visit the official MySQL website. If CDM fails to enable this function, contact the database administrator to enable the **local_infile** system variable. Alternatively, set **Use Local API** to **No** to disable API acceleration. If data is imported to RDS for MySQL, the LOAD DATA function is disabled by default. In such a case, you need to modify the parameter group of the MySQL instance and set **local_infile** to **ON** to enable the LOAD DATA function. **NOTE** If **local_infile** on RDS is uneditable, it is the default parameter group. You need to create a parameter group, modify its values, and apply it to the RDS for MySQL instance. For details, see the *Relational Database Service User Guide*.	Yes
Use Agent	Whether to extract data from the data source through an agent	Yes
Agent	Click **Select** and select the agent created in **Managing Agents**.	-
local_infile Character Set	When using local_infile to import data to MySQL, you can configure the encoding format.	utf8
Driver Version	Select a driver version that adapts to the database type.	-

Parameter	Description	Example Value
Fetch Size	(Optional) Displayed when you click **Show Advanced Attributes**. Number of rows obtained by each request. Set this parameter based on the data source and the job's data size. If the value is either too large or too small, the job may run for a long time.	1000
Commit Size	(Optional) Displayed when you click **Show Advanced Attributes**. Number of records submitted each time. Set this parameter based on the data destination and the job's data size. If the value is either too large or too small, the job may run for a long time.	-
SSL Encryption	(Optional) If you set this parameter to **Yes**, CDM can connect to the database (on-premises databases excluded) in SSL encryption mode.	Yes

Parameter	Description	Example Value
Link Attributes	(Optional) Click **Add** to add the JDBC connector attributes of multiple specified data sources. For details, see the JDBC connector document of the corresponding database. The following are some examples: ● **connectTimeout=600000** and **socketTimeout=300000**: When a large amount of data needs to be migrated or the entire table is retrieved using query statements, the migration fails due to connection timeout. In this case, you can customize the connection timeout interval (ms) and socket timeout interval (ms) to prevent failures caused by timeout. ● **tinyInt1isBit=false** or **mysql.bool.type.transform=false**: By default, **tinyInt1isBit** is **true**, indicating that **TINYINT(1)** is processed as a bit, that is, **Types.BOOLEAN**, and **1** or **0** is read as **true** or **false**. As a result, the migration fails. In this case, you can set **tinyInt1isBit** to **false** to avoid migration failures. ● **useCursorFetch=false**: By default, **useCursorFetch** is enabled, indicating that the JDBC connector communicates with relational databases using a binary protocol. Some third-party systems may have compatibility issues, causing migration time conversion errors. In this case, you can disable this function. Open-source MySQL databases support the **useCursorFetch** parameter, and you do not need to set this parameter. ● **allowPublicKeyRetrieval=true**: By default, public key retrieval is disabled for MySQL databases. If TLS is unavailable and an RSA public key is used for encryption, connection to an MySQL database may fail. In this case, you can enable public key retrieval to avoid connection failures.	sslmode=require
Reference Sign	(Optional) Delimiter between the names of the referenced tables or columns. For details, see the product documentation of the corresponding database.	`

Parameter	Description	Example Value
Batch Size	Number of rows written each time. It should be less than Commit Size. When the number of rows written reaches the value of Commit Size, the rows will be committed to the database.	100

4.9 Link to an Oracle Database

Table 4-8 lists the parameters for a link to an Oracle database.

Table 4-8 Oracle database link parameters

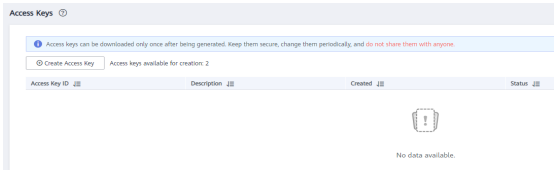
Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	oracle_link
Database Server	IP address or domain name of the database to connect	192.168.0.1
Port	Port of the database to connect	Default port: 1521
Connection Type	Oracle database connection type. The following options are available: • **Service Name**: Use **SERVICE_NAME** to connect to the Oracle database. • **SID**: Use **SID** to connect to the Oracle database.	SID
Instance Name	Oracle instance ID, which is used to differentiate databases by instances. This parameter is available only when **Connection Type** is set to **SID**.	dbname
Database Name	Name of the database to connect This parameter is available only when **Connection Type** is set to **Service Name**.	dbname
Username	Username used for accessing the database This account must have the permissions required to read and write data tables and metadata.	cdm
Password	Password of the username	-
Use Agent	Whether to extract data from the data source through an agent	Yes
Agent	Click **Select** and select the agent created in **Managing Agents**.	-

Parameter	Description	Example Value
Oracle Version	Oracle database version. This parameter is available only for Oracle links. If **java.sql.SQLException: Protocol violation** is displayed, select another version.	Later than 12.1
Fetch Size	(Optional) Displayed when you click **Show Advanced Attributes**. Number of rows obtained by each request. Set this parameter based on the data source and the job's data size. If the value is either too large or too small, the job may run for a long time. A migration from the Oracle to DWS database may time out due to a long data write duration in the DWS database. In this case, reduce the value of **Fetch Size** for the Oracle database.	1000
Link Attributes	(Optional) Click **Add** to add the JDBC connector attributes of multiple specified data sources. For details, see the JDBC connector document of the corresponding database. The following are some examples: • **oracle.net.CONNECT_TIMEOUT=60000** and **oracle.jdbc.ReadTimeout=300000**: When a large amount of data needs to be migrated or the entire table is retrieved using query statements, the migration fails due to connection timeout. In this case, you can customize the connection timeout interval (ms) and read timeout interval (ms) to prevent failures caused by timeout.	sslmode=require
Reference Sign	(Optional) Delimiter between the names of the referenced tables or columns. For details, see the product documentation of the corresponding database.	"
Driver Version	Different types of relational databases adapt to different drivers. For details, see **How Do I Obtain a Driver?**	-

4.10 Link to DLI

When connecting CDM to DLI, configure the parameters as described in **Table 4-9**.

Table 4-9 DLI link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	dli_link
AK	AK/SK required for authentication during access to the DLI database. You need to create an access key for the current account and obtain an AK/SK pair. 1. Log in to the management console, move the cursor to the username in the upper right corner, and select **My Credentials** from the drop-down list. 2. On the **My Credentials** page, choose **Access Keys**, and click **Create Access Key**. See **Figure 4-13**. **Figure 4-13** Clicking Create Access Key  3. Click **OK** and save the access key file as prompted. The access key file will be saved to your browser's configured download location. Open the **credentials.csv** file to view **Access Key Id** and **Secret Access Key**. **NOTE** • Only two access keys can be added for each user. • To ensure access key security, the access key is automatically downloaded only when it is generated for the first time and cannot be obtained from the management console later. Keep them properly.	-
SK		-

Parameter	Description	Example Value
Project ID	Project ID in the region where DLI resides A project is a group of tenant resources, and an account ID corresponds to the current account. The IAM ID corresponds to the current user. You can view the project IDs, account IDs, and user IDs in different regions on the corresponding pages. 1. Register with and log in to the management console. 2. Hover the cursor on the username in the upper right corner and select **My Credentials** from the drop-down list. 3. On the **API Credentials** page, obtain the account name, account ID, IAM username, and IAM user ID, and obtain the project ID from the project list.	-

4.11 Link to Hive

CDM supports the following Hive data sources:

- **MRS Hive**
- **FusionInsight Hive**
- **Apache Hive**

MRS Hive

You can view a table during field mapping only when you have the permission to access the table connected to MRS Hive.

MRS Hive links apply to the MapReduce Service (MRS) on Huawei Cloud. **Table 4-10** describes related parameters.

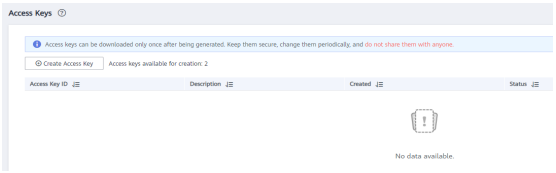
NOTE

- Before creating an MRS Hive link, you need to add an authenticated Kerberos user on MRS and log in to the MRS management page to change the initial password. Then use the new user to create an MRS link.
- To connect to an MRS 2.*x* cluster, create a CDM cluster of version 2.*x* first. CDM 1.8.*x* clusters cannot connect to MRS 2.*x* clusters.
- Currently, the Hive link obtains the **core-site.xml** configuration information from MRS HDFS. Therefore, if MRS Hive uses OBS as the underlying storage system, configure the AK/SK of OBS on MRS HDFS before creating the Hive link.
- Ensure that the MRS cluster and the DataArts Studio instance can communicate with each other. The following requirements must be met for network interconnection:
  - If the CDM cluster in the DataArts Studio instance and the MRS cluster are in different regions, a public network or a dedicated connection is required. If the Internet is used for communication, ensure that an EIP has been bound to the CDM cluster, and the MRS cluster can access the Internet and the port has been enabled in the firewall rule.
  - If the CDM cluster in the DataArts Studio instance and the cloud service are in the same region, VPC, subnet, and security group, they can communicate with each other by default. If they are in the same VPC but in different subnets or security groups, you must configure routing rules and security group rules. For details about how to configure routing rules, see **configuring routes**. For details about how to configure security group rules, see **configuring security group rules**.
  - The MRS cluster and the DataArts Studio workspace belong to the same enterprise project. If they do not, you can modify the enterprise project of the workspace.

Table 4-10 MRS Hive link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	hivelink
Manager IP	Floating IP address of MRS Manager. Click **Select** next to the **Manager IP** text box to select an MRS cluster. CDM automatically fills in the authentication information.	127.0.0.1
Authentication Method	Authentication method used for accessing MRS • **SIMPLE**: Select this for non-security mode. • **KERBEROS**: Select this for security mode.	SIMPLE
HIVE Version	Set this to the Hive version on the server.	HIVE_3_X

Parameter	Description	Example Value
Username	If **Authentication Method** is set to **KERBEROS**, you must provide the username and password used for logging in to MRS Manager. If you need to create a snapshot when exporting a directory from HDFS, the user configured here must have the administrator permission on HDFS. To create a data connection for an MRS security cluster, do not use user **admin**. The **admin** user is the default management page user and cannot be used as the authentication user of the security cluster. You can create an MRS user and set **Username** and **Password** to the username and password of the created MRS user when creating an MRS data connection. **NOTE** • If the CDM cluster version is 2.9.0 or later and the MRS cluster version is 3.1.0 or later, the created user must have the permissions of the **Manager_viewer** role to create links on CDM. To perform operations on databases, tables, and columns of an MRS component, you also need to add the database, table, and column permissions of the MRS component to the user by following the instructions in the MRS documentation. • If the CDM cluster version is earlier than 2.9.0 or the MRS cluster version is earlier than 3.1.0, the created user must have the permissions of **Manager_administrator** or **System_administrator** to create links on CDM. • A user with only the **Manager_tenant** or **Manager_auditor** permission cannot create connections.	cdm
Password	Password used for logging in to MRS Manager	-
Enable ldap	This parameter is available when **Proxy connection** is selected for **Connection Type**. If LDAP authentication is enabled for an external LDAP server connected to MRS Hive, the LDAP username and password are required for authenticating the connection to MRS Hive. In this case, this option must be enabled. Otherwise, the connection will fail.	No
ldapUsername	This parameter is mandatory when **Enable ldap** is enabled. Enter the username configured when LDAP authentication was enabled for MRS Hive.	-

Parameter	Description	Example Value
ldapPassword	This parameter is mandatory when **Enable ldap** is enabled. Enter the password configured when LDAP authentication was enabled for MRS Hive.	-
OBS storage support	The server must support OBS storage. When creating a Hive table, you can store the table in OBS.	No
AK	This parameter is mandatory when **OBS storage support** is enabled. The account corresponding to the AK/SK pair must have the OBS Buckets Viewer permission. Otherwise, OBS cannot be accessed and the "403 AccessDenied" error is reported. You need to create an access key for the current account and obtain an AK/SK pair. 1. Log in to the management console, move the cursor to the username in the upper right corner, and select **My Credentials** from the drop-down list. 2. On the **My Credentials** page, choose **Access Keys**, and click **Create Access Key**. See **Figure 4-14**. **Figure 4-14** Clicking Create Access Key  3. Click **OK** and save the access key file as prompted. The access key file will be saved to your browser's configured download location. Open the **credentials.csv** file to view **Access Key Id** and **Secret Access Key**. **NOTE** • Only two access keys can be added for each user. • To ensure access key security, the access key is automatically downloaded only when it is generated for the first time and cannot be obtained from the management console later. Keep them properly.	-
SK		-

Parameter	Description	Example Value
Run Mode	This parameter is used only when the Hive version is **HIVE_3_X**. Possible values are: • **EMBEDDED**: The link instance runs with CDM. This mode delivers better performance. • **Standalone**: The link instance runs in an independent process. If CDM needs to connect to multiple Hadoop data sources (MRS, Hadoop, or CloudTable) with both Kerberos and Simple authentication modes, **Standalone** prevails. **NOTE** The **STANDALONE** mode is used to solve the version conflict problem. If the connector versions of the source and destination ends of the same link are different, a JAR file conflict occurs. In this case, you need to place the source or destination end in the STANDALONE process to prevent the migration failure caused by the conflict.	EMBEDDED
Check Hive JDBC Connectivity	Whether to check the Hive JDBC connectivity	No
Use Cluster Config	You can use the cluster configuration to simplify parameter settings for the Hadoop connection.	No
Cluster Config Name	This parameter is valid only when **Use Cluster Config** is set to **Yes**. Select a cluster configuration that has been created. For details about how to configure a cluster, see **Managing Cluster Configurations**.	hive_01

Click **Show Advanced Attributes**, and then click **Add** to add configuration attributes of other clients. The name and value of each attribute must be configured. You can click **Delete** to delete no longer used attributes.

The following are some examples:

- **connectTimeout=360000** and **socketTimeout=360000**: When a large amount of data needs to be migrated or the entire table is retrieved using query statements, the migration fails due to connection timeout. In this case, you can customize the connection timeout interval (ms) and socket timeout interval (ms) to prevent failures caused by timeout.
- **hive.server2.idle.operation.timeout=360000**: To prevent Hive migration jobs from being suspended for a long time, you can customize the operation timeout period (ms).
- **hive.storeFormat=textfile**: During data migration from a relational database to Hive, tables in ORC format are automatically created by default. If you

want textfile or parquet tables to be created, add **hive.storeFormat=textfile** or **hive.storeFormat=parquet**.

- **fs.defaultFS=obs://hivedb**: If the interconnected MRS Hive uses decoupled storage and compute, you can use this configuration to achieve better compatibility.

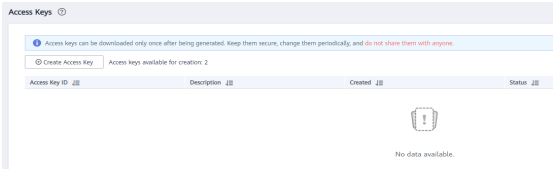
FusionInsight Hive

The FusionInsight Hive link is applicable to data migration of FusionInsight HD in the local data center. You must use Direct Connect to connect to FusionInsight HD.

Table 4-11 describes related parameters.

Table 4-11 FusionInsight Hive link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	hivelink
Manager IP	IP address of FusionInsight Manager	127.0.0.1
Manager Port	Port number of FusionInsight Manager	28443
CAS Server Port	Port number of the CAS server used to connect to FusionInsight	20009
Authentication Method	Authentication method used for accessing the cluster: • **SIMPLE**: Select this for non-security mode. • **KERBEROS**: Select this for security mode.	SIMPLE
HIVE Version	Hive version	HIVE_3_X
Username	Username used for logging in to FusionInsight Manager.	cdm
Password	Password used for logging in to FusionInsight Manager	-
OBS storage support	The server must support OBS storage. When creating a Hive table, you can store the table in OBS.	No

Parameter	Description	Example Value
AK	This parameter is mandatory when **OBS storage support** is enabled. The account corresponding to the AK/SK pair must have the OBS Buckets Viewer permission. Otherwise, OBS cannot be accessed and the "403 AccessDenied" error is reported. You need to create an access key for the current account and obtain an AK/SK pair. 1. Log in to the management console, move the cursor to the username in the upper right corner, and select **My Credentials** from the drop-down list. 2. On the **My Credentials** page, choose **Access Keys**, and click **Create Access Key**. See **Figure 4-15**. **Figure 4-15** Clicking Create Access Key  3. Click **OK** and save the access key file as prompted. The access key file will be saved to your browser's configured download location. Open the **credentials.csv** file to view **Access Key Id** and **Secret Access Key**. **NOTE** • Only two access keys can be added for each user. • To ensure access key security, the access key is automatically downloaded only when it is generated for the first time and cannot be obtained from the management console later. Keep them properly.	-
SK		-

Parameter	Description	Example Value
Run Mode	This parameter is used only when the Hive version is **HIVE_3_X**. Possible values are: • **EMBEDDED**: The link instance runs with CDM. This mode delivers better performance. • **Standalone**: The link instance runs in an independent process. If CDM needs to connect to multiple Hadoop data sources (MRS, Hadoop, or CloudTable) with both Kerberos and Simple authentication modes, **Standalone** prevails. **NOTE** The **STANDALONE** mode is used to solve the version conflict problem. If the connector versions of the source and destination ends of the same link are different, a JAR file conflict occurs. In this case, you need to place the source or destination end in the STANDALONE process to prevent the migration failure caused by the conflict.	EMBEDDED
Use Cluster Config	You can use the cluster configuration to simplify parameter settings for the Hadoop connection.	No
Cluster Config Name	This parameter is valid only when **Use Cluster Config** is set to **Yes**. Select a cluster configuration that has been created. For details about how to configure a cluster, see **Managing Cluster Configurations**.	hive_01

Click **Show Advanced Attributes**, and then click **Add** to add configuration attributes of other clients. The name and value of each attribute must be configured. You can click **Delete** to delete no longer used attributes.

The following are some examples:

- **connectTimeout=360000** and **socketTimeout=360000**: When a large amount of data needs to be migrated or the entire table is retrieved using query statements, the migration fails due to connection timeout. In this case, you can customize the connection timeout interval (ms) and socket timeout interval (ms) to prevent failures caused by timeout.
- **hive.server2.idle.operation.timeout=360000**: To prevent Hive migration jobs from being suspended for a long time, you can customize the operation timeout period (ms).

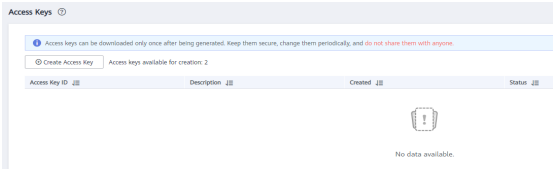
Apache Hive

The Apache Hive link is applicable to data migration of the third-party Hadoop in the local data center or ECS. You must use Direct Connect to connect to Hadoop in the local data center.

Table 4-12 describes related parameters.

Table 4-12 Apache Hive link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	hivelink
URI	NameNode URI	hdfs://hacluster
Hive Metastore	Hive metadata address. For details, see the **hive.metastore.uris** configuration item. Example: thrift://host-192-168-1-212:9083	-
Authentication Method	Authentication method used for accessing the cluster: • **SIMPLE**: Select this for non-security mode. • **KERBEROS**: Select this for security mode.	SIMPLE
Hive Version	Hive version	HIVE_3_X
IP and Host Name Mapping	If the Hadoop configuration file uses the host name, configure the mapping between the IP address and host name. Separate the IP addresses and host names by spaces and mappings by semicolons (;), carriage returns, or line feeds.	-
OBS storage support	The server must support OBS storage. When creating a Hive table, you can store the table in OBS.	No

Parameter	Description	Example Value
AK	This parameter is mandatory when **OBS storage support** is enabled. The account corresponding to the AK/SK pair must have the OBS Buckets Viewer permission. Otherwise, OBS cannot be accessed and the "403 AccessDenied" error is reported. You need to create an access key for the current account and obtain an AK/SK pair. 1. Log in to the management console, move the cursor to the username in the upper right corner, and select **My Credentials** from the drop-down list. 2. On the **My Credentials** page, choose **Access Keys**, and click **Create Access Key**. See **Figure 4-16**. **Figure 4-16** Clicking Create Access Key  3. Click **OK** and save the access key file as prompted. The access key file will be saved to your browser's configured download location. Open the **credentials.csv** file to view **Access Key Id** and **Secret Access Key**. **NOTE** • Only two access keys can be added for each user. • To ensure access key security, the access key is automatically downloaded only when it is generated for the first time and cannot be obtained from the management console later. Keep them properly.	-
SK		-

Parameter	Description	Example Value
Run Mode	This parameter is used only when the Hive version is **HIVE_3_X**. Possible values are: • **EMBEDDED**: The link instance runs with CDM. This mode delivers better performance. • **Standalone**: The link instance runs in an independent process. If CDM needs to connect to multiple Hadoop data sources (MRS, Hadoop, or CloudTable) with both Kerberos and Simple authentication modes, **Standalone** prevails. **NOTE** The **STANDALONE** mode is used to solve the version conflict problem. If the connector versions of the source and destination ends of the same link are different, a JAR file conflict occurs. In this case, you need to place the source or destination end in the STANDALONE process to prevent the migration failure caused by the conflict.	EMBEDDED
Use Cluster Config	You can use the cluster configuration to simplify parameter settings for the Hadoop connection.	No
Cluster Config Name	This parameter is valid when **Use Cluster Config** is set to **Yes** or **Authentication Method** is set to **KERBEROS**. Select a cluster configuration that has been created. For details about how to configure a cluster, see **Managing Cluster Configurations**.	hive_01
Hive JDBC URL	URL for connecting to Hive JDBC. By default, anonymous users are used.	-

Click **Show Advanced Attributes**, and then click **Add** to add configuration attributes of other clients. The name and value of each attribute must be configured. You can click **Delete** to delete no longer used attributes.

The following are some examples:

- **connectTimeout=360000** and **socketTimeout=360000**: When a large amount of data needs to be migrated or the entire table is retrieved using query statements, the migration fails due to connection timeout. In this case, you can customize the connection timeout interval (ms) and socket timeout interval (ms) to prevent failures caused by timeout.
- **hive.server2.idle.operation.timeout=360000**: To prevent Hive migration jobs from being suspended for a long time, you can customize the operation timeout period (ms).

4.12 Link to HBase

CDM supports the following HBase data sources:

- **MRS HBase**
- **FusionInsight HBase**
- **Apache HBase**

MRS HBase

When connecting CDM to HBase of MRS, configure the parameters as described in **Table 4-13**.

NOTE

- Before creating an MRS link, you need to add an authenticated Kerberos user on MRS and log in to the MRS management page to change the initial password. Then use the new user to create an MRS link.
- To connect to an MRS 2.*x* cluster, create a CDM cluster of version 2.*x* first. CDM 1.8.*x* clusters cannot connect to MRS 2.*x* clusters.
- If the connection fails after you select a cluster, check whether the MRS cluster can communicate with the CDM instance which functions as the agent. They can communicate with each other in the following scenarios:
  - If the CDM cluster in the DataArts Studio instance and the MRS cluster are in different regions, a public network or a dedicated connection is required. If the Internet is used for communication, ensure that an EIP has been bound to the CDM cluster, and the MRS cluster can access the Internet and the port has been enabled in the firewall rule.
  - If the CDM cluster in the DataArts Studio instance and the cloud service are in the same region, VPC, subnet, and security group, they can communicate with each other by default. If they are in the same VPC but in different subnets or security groups, you must configure routing rules and security group rules. For details about how to configure routing rules, see **configuring routes**. For details about how to configure security group rules, see **configuring security group rules**.
  - The MRS cluster and the DataArts Studio workspace belong to the same enterprise project. If they do not, you can modify the enterprise project of the workspace.

Table 4-13 MRS HBase link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	mrs_hbase_link
Manager IP	Floating IP address of MRS Manager. Click **Select** next to the **Manager IP** text box to select an MRS cluster. CDM automatically fills in the authentication information.	127.0.0.1

Parameter	Description	Example Value
Username	If **Authentication Method** is set to **KERBEROS**, you must provide the username and password used for logging in to MRS Manager. If you need to create a snapshot when exporting a directory from HDFS, the user configured here must have the administrator permission on HDFS. To create a data connection for an MRS security cluster, do not use user **admin**. The **admin** user is the default management page user and cannot be used as the authentication user of the security cluster. You can create an MRS user and set **Username** and **Password** to the username and password of the created MRS user when creating an MRS data connection. **NOTE** • If the CDM cluster version is 2.9.0 or later and the MRS cluster version is 3.1.0 or later, the created user must have the permissions of the **Manager_viewer** role to create links on CDM. To perform operations on databases, tables, and columns of an MRS component, you also need to add the database, table, and column permissions of the MRS component to the user by following the instructions in the MRS documentation. • If the CDM cluster version is earlier than 2.9.0 or the MRS cluster version is earlier than 3.1.0, the created user must have the permissions of **Manager_administrator** or **System_administrator** to create links on CDM. • A user with only the **Manager_tenant** or **Manager_auditor** permission cannot create connections.	cdm
Password	Password used for logging in to MRS Manager	-
Authentication Method	Authentication method used for accessing the cluster: • **SIMPLE**: Select this for non-security mode. • **KERBEROS**: Select this for security mode.	SIMPLE
HBase Version	HBase version	HBASE_2_X

Parameter	Description	Example Value
Run Mode	Run mode of the HBase link. This parameter is used only when the HBase version is **HBASE_2_X**. • **EMBEDDED**: The link instance runs with CDM. This mode delivers better performance. • **Standalone**: The link instance runs in an independent process. If CDM needs to connect to multiple Hadoop data sources (MRS, Hadoop, or CloudTable) with both Kerberos and Simple authentication modes, **Standalone** prevails. Note: The STANDALONE mode is used to solve the version conflict problem. If the connector versions of the source and destination ends of the same link are different, a JAR file conflict occurs. In this case, you need to place the source or destination end in the STANDALONE process to prevent the migration failure caused by the conflict.	STANDALONE
Use Cluster Config	You can create cluster configurations on the **Links** page to simplify the configuration of Hadoop link parameters.	No
Cluster Config Name	This parameter is valid only when **Use Cluster Config** is set to **Yes**. Select a cluster configuration that has been created. For details about how to configure a cluster, see **Managing Cluster Configurations**.	hbase_01

Click **Show Advanced Attributes**, and then click **Add** to add configuration attributes of other clients. The name and value of each attribute must be configured. You can click **Delete** to delete no longer used attributes.

FusionInsight HBase

When connecting CDM to HBase of FusionInsight HD, configure the parameters as described in **Table 4-14**.

Table 4-14 FusionInsight HBase link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	FI_hbase_link
Manager IP	IP address of FusionInsight Manager	127.0.0.1
Manager Port	Port number of FusionInsight Manager	28443
CAS Server Port	Port number of the CAS server used to connect to FusionInsight	20009
Username	Username used for logging in to FusionInsight Manager.	cdm
Password	Password used for logging in to FusionInsight Manager	-
Authentication Method	Authentication method used for accessing the cluster: • **SIMPLE**: Select this for non-security mode. • **KERBEROS**: Select this for security mode.	Kerberos
HBase Version	HBase version	HBASE_2_X
Run Mode	Run mode of the HBase link. This parameter is used only when the HBase version is **HBASE_2_X**. • **EMBEDDED**: The link instance runs with CDM. This mode delivers better performance. • **Standalone**: The link instance runs in an independent process. If CDM needs to connect to multiple Hadoop data sources (MRS, Hadoop, or CloudTable) with both Kerberos and Simple authentication modes, **Standalone** prevails. **NOTE** The **STANDALONE** mode is used to solve the version conflict problem. If the connector versions of the source and destination ends of the same link are different, a JAR file conflict occurs. In this case, you need to place the source or destination end in the STANDALONE process to prevent the migration failure caused by the conflict.	STANDALONE
Use Cluster Config	You can use the cluster configuration to simplify parameter settings for the Hadoop connection.	No

Parameter	Description	Example Value
Cluster Config Name	This parameter is valid only when **Use Cluster Config** is set to **Yes**. Select a cluster configuration that has been created. For details about how to configure a cluster, see **Managing Cluster Configurations**.	hbase_01

Click **Show Advanced Attributes**, and then click **Add** to add configuration attributes of other clients. The name and value of each attribute must be configured. You can click **Delete** to delete no longer used attributes.

Apache HBase

When connecting CDM to HBase of Apache Hadoop, configure the parameters as described in **Table 4-15**.

Table 4-15 Apache HBase link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	hadoop_hbase_link
ZK Link	ZooKeeper link of HBase Format: <host1>:<port>,<host2>:<port>,<host3>:<port>	zk1.example.com:2181,zk2.example.com:2181,zk3.example.com:2181
Authentication Method	Authentication method used for accessing the cluster: • **SIMPLE**: Select this for non-security mode. • **KERBEROS**: Select this for security mode.	Kerberos
IP and Host Name Mapping	IP address and host name. If the configuration file uses host names, configure the mappings between all IP addresses and hosts. Use spaces to separate hosts.	IP: 10.3.6.9 Host name: hostname01
HBase Version	HBase version	HBASE_2_X

Parameter	Description	Example Value
Run Mode	Run mode of the HBase link. This parameter is used only when the HBase version is **HBASE_2_X**. • **EMBEDDED**: The link instance runs with CDM. This mode delivers better performance. • **Standalone**: The link instance runs in an independent process. If CDM needs to connect to multiple Hadoop data sources (MRS, Hadoop, or CloudTable) with both Kerberos and Simple authentication modes, **Standalone** prevails. **NOTE** The **STANDALONE** mode is used to solve the version conflict problem. If the connector versions of the source and destination ends of the same link are different, a JAR file conflict occurs. In this case, you need to place the source or destination end in the STANDALONE process to prevent the migration failure caused by the conflict.	STANDALONE
Use Cluster Config	You can use the cluster configuration to simplify parameter settings for the Hadoop connection.	No
Cluster Config Name	This parameter is valid only when **Use Cluster Config** is set to **Yes**. Select a cluster configuration that has been created. For details about how to configure a cluster, see **Managing Cluster Configurations**.	hbase_01

Click **Show Advanced Attributes**, and then click **Add** to add configuration attributes of other clients. The name and value of each attribute must be configured. You can click **Delete** to delete no longer used attributes.

4.13 Link to HDFS

CDM supports the following HDFS data sources:

- **MRS HDFS**
- **FusionInsight HDFS**
- **Apache HDFS**

MRS HDFS

When connecting CDM to HDFS of MRS, configure the parameters as described in **Table 4-16**.

NOTE

- Before creating an MRS link, you need to add an authenticated Kerberos user on MRS and log in to the MRS management page to change the initial password. Then use the new user to create an MRS link.
- To connect to an MRS 2.*x* cluster, create a CDM cluster of version 2.*x* first. CDM 1.8.*x* clusters cannot connect to MRS 2.*x* clusters.
- If the connection fails after you select a cluster, check whether the MRS cluster can communicate with the CDM instance which functions as the agent. They can communicate with each other in the following scenarios:
  - If the CDM cluster in the DataArts Studio instance and the MRS cluster are in different regions, a public network or a dedicated connection is required. If the Internet is used for communication, ensure that an EIP has been bound to the CDM cluster, and the MRS cluster can access the Internet and the port has been enabled in the firewall rule.
  - If the CDM cluster in the DataArts Studio instance and the cloud service are in the same region, VPC, subnet, and security group, they can communicate with each other by default. If they are in the same VPC but in different subnets or security groups, you must configure routing rules and security group rules. For details about how to configure routing rules, see **configuring routes**. For details about how to configure security group rules, see **configuring security group rules**.
  - The MRS cluster and the DataArts Studio workspace belong to the same enterprise project. If they do not, you can modify the enterprise project of the workspace.

Table 4-16 MRS HDFS link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	mrs_hdfs_link
Manager IP	Floating IP address of MRS Manager. Click **Select** next to the **Manager IP** text box to select an MRS cluster. CDM automatically fills in the authentication information.	127.0.0.1

Parameter	Description	Example Value
Username	If **Authentication Method** is set to **KERBEROS**, you must provide the username and password used for logging in to MRS Manager. If you need to create a snapshot when exporting a directory from HDFS, the user configured here must have the administrator permission on HDFS. To create a data connection for an MRS security cluster, do not use user **admin**. The **admin** user is the default management page user and cannot be used as the authentication user of the security cluster. You can create an MRS user and set **Username** and **Password** to the username and password of the created MRS user when creating an MRS data connection. **NOTE** • If the CDM cluster version is 2.9.0 or later and the MRS cluster version is 3.1.0 or later, the created user must have the permissions of the **Manager_viewer** role to create links on CDM. To perform operations on databases, tables, and columns of an MRS component, you also need to add the database, table, and column permissions of the MRS component to the user by following the instructions in the MRS documentation. • If the CDM cluster version is earlier than 2.9.0 or the MRS cluster version is earlier than 3.1.0, the created user must have the permissions of **Manager_administrator** or **System_administrator** to create links on CDM. • A user with only the **Manager_tenant** or **Manager_auditor** permission cannot create connections.	cdm
Password	Password used for logging in to MRS Manager	-
Authentication Method	Authentication method used for accessing MRS • **SIMPLE**: Select this for non-security mode. • **KERBEROS**: Select this for security mode.	SIMPLE

Parameter	Description	Example Value
Run Mode	Run mode of the HDFS link. The options are as follows: • **EMBEDDED**: The link instance runs with CDM. This mode delivers better performance. • **STANDALONE**: The link instance runs in an independent process. If CDM needs to connect to multiple Hadoop data sources (MRS, Hadoop, or CloudTable) with both Kerberos and Simple authentication modes, select **STANDALONE** or configure different agents. Note: The STANDALONE mode is used to solve the version conflict problem. If the connector versions of the source and destination ends of the same link are different, a JAR file conflict occurs. In this case, you need to place the source or destination end in the STANDALONE process to prevent the migration failure caused by the conflict. • **Agent**: The link instance runs on an agent. If **Agent** is not used, and the CDM cluster connects to two or more clusters with Kerberos authentication enabled and the same realm, only one cluster can be connected in **EMBEDDED** mode, and the other clusters must be in **STANDALONE** mode.	STANDALONE
Agent	Click **Select** and select the agent created in **Connecting to an Agent**. This parameter is displayed when **Run Mode** is set to **Agent**.	-
Use Cluster Config	You can use the cluster configuration to simplify parameter settings for the Hadoop connection.	No
Cluster Config Name	This parameter is valid only when **Use Cluster Config** is set to **Yes**. Select a cluster configuration that has been created. For details about how to configure a cluster, see **Managing Cluster Configurations**.	hdfs_01

Click **Show Advanced Attributes**, and then click **Add** to add configuration attributes of other clients. The name and value of each attribute must be configured. You can click **Delete** to delete no longer used attributes.

FusionInsight HDFS

When connecting CDM to HDFS of FusionInsight HD, configure the parameters as described in **Table 4-17**.

Table 4-17 FusionInsight HDFS link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	FI_hdfs_link
Manager IP	IP address of FusionInsight Manager	127.0.0.1
Manager Port	Port number of FusionInsight Manager	28443
CAS Server Port	Port number of the CAS server used to connect to FusionInsight	20009
Username	Username used for logging in to FusionInsight Manager. If you need to create a snapshot when exporting a directory from HDFS, the user configured here must have the administrator permission on HDFS.	cdm
Password	Password used for logging in to FusionInsight Manager	-
Authentication Method	Authentication method used for accessing the cluster: • **SIMPLE**: Select this for non-security mode. • **KERBEROS**: Select this for security mode.	KERBEROS

Parameter	Description	Example Value
Run Mode	Run mode of the HDFS link. The options are as follows: • **EMBEDDED**: The link instance runs with CDM. This mode delivers better performance. • **STANDALONE**: The link instance runs in an independent process. If CDM needs to connect to multiple Hadoop data sources (MRS, Hadoop, or CloudTable) with both Kerberos and Simple authentication modes, select **STANDALONE** or configure different agents. Note: The STANDALONE mode is used to solve the version conflict problem. If the connector versions of the source and destination ends of the same link are different, a JAR file conflict occurs. In this case, you need to place the source or destination end in the STANDALONE process to prevent the migration failure caused by the conflict. • **Agent**: The link instance runs on an agent.	STANDALONE
Agent	Click **Select** and select the agent created in **Connecting to an Agent**. This parameter is displayed when **Run Mode** is set to **Agent**.	-
Use Cluster Config	You can use the cluster configuration to simplify parameter settings for the Hadoop connection.	No
Cluster Config Name	This parameter is valid only when **Use Cluster Config** is set to **Yes**. Select a cluster configuration that has been created. For details about how to configure a cluster, see **Managing Cluster Configurations**.	hdfs_01

Click **Show Advanced Attributes**, and then click **Add** to add configuration attributes of other clients. The name and value of each attribute must be configured. You can click **Delete** to delete no longer used attributes.

Apache HDFS

When connecting CDM to HDFS of Apache Hadoop, configure the parameters as described in **Table 4-18**.

Table 4-18 Apache HDFS link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	hadoop_hdfs_link
URI	NameNode URI You can enter **hdfs://*IP address of the NameNode instance*:8020**.	hdfs://*IP*:8020
Authentication Method	Authentication method used for accessing the cluster: ● **SIMPLE**: Select this for non-security mode. ● **KERBEROS**: Select this for security mode.	KERBEROS
Run Mode	Run mode of the HDFS link. The options are as follows: ● **EMBEDDED**: The link instance runs with CDM. This mode delivers better performance. ● **STANDALONE**: The link instance runs in an independent process. If CDM needs to connect to multiple Hadoop data sources (MRS, Hadoop, or CloudTable) with both Kerberos and Simple authentication modes, select **STANDALONE** or configure different agents. Note: The STANDALONE mode is used to solve the version conflict problem. If the connector versions of the source and destination ends of the same link are different, a JAR file conflict occurs. In this case, you need to place the source or destination end in the STANDALONE process to prevent the migration failure caused by the conflict. ● **Agent**: The link instance runs on an agent. For Apache HDFS, you can select **Agent** only if **Authentication Method** is set to **SIMPLE**.	STANDALONE
IP and Host Name Mapping	This parameter is used only when **Run Mode** is set to **EMBEDDED** or **STANDALONE**. If the HDFS configuration file uses the host name, configure the mapping between the IP address and host name. Separate the IP addresses and host names by spaces and mappings by semicolons (;), carriage returns, or line feeds.	10.1.6.9 hostname01 10.2.7.9 hostname02

Parameter	Description	Example Value
Agent	This parameter is required when **Authentication Method** is set to **SIMPLE** and **Run Mode** is set to **Agent**. Select the agent created in **Connecting to an Agent**.	-
Use Cluster Config	You can use the cluster configuration to simplify parameter settings for the Hadoop connection.	No
Cluster Config Name	This parameter is valid when **Use Cluster Config** is set to **Yes** or **Authentication Method** is set to **KERBEROS**. Select a cluster configuration that has been created. For details about how to configure a cluster, see **Managing Cluster Configurations**.	hdfs_01

4.14 Link to an FTP or SFTP Server

The FTP/SFTP link is used to migrate files from the on-premises file server or ECS to a database.

NOTE

Only FTP servers running Linux are supported.

When connecting CDM to an FTP or SFTP server, configure the parameters as described in **Table 4-19**.

Table 4-19 FTP/SFTP link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	ftp_link
Host Name/IP Address	Host name or IP address of the FTP or SFTP server	ftp.apache.org
Port	Port number of the FTP or SFTP server. The default value is **21** for FTP and **22** for SFTP.	21
Username	Username used for logging in to the FTP or SFTP server	cdm
Password	Password used for logging in to the FTP or SFTP server	-

4.15 Link to Redis

The Redis link is applicable to data migration of Redis created in the local data center or ECS. It is used to load data in the database or files to Redis.

Links to Redis data encrypted using SSL are not supported.

When connecting CDM to an on-premises Redis database, configure the parameters as described in **Table 4-20**.

Table 4-20 Redis link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	redis_link
Redis Deployment Method	Two deployment methods are available: • **Single**: installation on a single-node system • **Cluster**: installation on a cluster • **Proxy**: installation using a proxy	Single
Redis Server List	List of Redis server addresses. Enter each address in the format of *IP address or domain name of the database server*:*port number*, Separate multiple server lists by semicolons (;).	192.168.0.1:7300;192.168.0.2:7301
Password	Password used for logging in to Redis	-
Redis Database Index	Index ID of a Redis database A Redis database is similar to a relational database. The total number of Redis databases can be set in the Redis configuration file. By default, there are 16 Redis databases. The database names are integers ranging from 0 to 15 instead of character strings.	0
Authentication Method	Authentication method used for accessing MRS • **SIMPLE**: Select this for non-security mode. • **KERBEROS**: Select this for security mode.	SIMPLE

Parameter	Description	Example Value
Username	If **Authentication Method** is set to **KERBEROS**, you must provide the username and password used for logging in to MRS Manager. If you need to create a snapshot when exporting a directory from HDFS, the user configured here must have the administrator permission on HDFS. To create a data connection for an MRS security cluster, do not use user **admin**. The **admin** user is the default management page user and cannot be used as the authentication user of the security cluster. You can create an MRS user and set **Username** and **Password** to the username and password of the created MRS user when creating an MRS data connection. **NOTE** • If the CDM cluster version is 2.9.0 or later and the MRS cluster version is 3.1.0 or later, the created user must have the permissions of the **Manager_viewer** role to create links on CDM. To perform operations on databases, tables, and columns of an MRS component, you also need to add the database, table, and column permissions of the MRS component to the user by following the instructions in the MRS documentation. • If the CDM cluster version is earlier than 2.9.0 or the MRS cluster version is earlier than 3.1.0, the created user must have the permissions of **Manager_administrator** or **System_administrator** to create links on CDM. • A user with only the **Manager_tenant** or **Manager_auditor** permission cannot create connections.	cdm
Cluster Config Name	This parameter is valid only when **Authentication Method** is set to **KERBEROS**. Select a cluster configuration you have created. For details about how to configure a cluster, see **Managing Cluster Configurations**.	hdfs_01

4.16 Link to DDS

The DDS link is used to synchronize data from Document Database Service (DDS) on HUAWEI CLOUD to a big data platform.

When connecting CDM to DDS, configure the parameters as described in **Table 4-21**.

Table 4-21 DDS link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	dds_link
Server List	List of server addresses. Enter each address in the format of *IP address or domain name of the database server*:*port number*, and separate the entered addresses with semicolons (;).	192.168.0.1:7300;192.168.0.2:7301
Database Name	Name of the DDS database to be connected	DB_dds
Username	Username used for logging in to DDS	cdm
Password	Password used for logging in to DDS	-
Is direct connection mode	This mode applies to the scenario where the network of the primary node is normal but that of the replica node is abnormal. **NOTE** • Only one IP address can be configured for the server list in direct connection mode. • This mode applies to the scenario where the network of the primary node is normal but the network of the replica node is abnormal.	No

4.17 Link to CloudTable

When connecting CDM to CloudTable, configure the parameters as described in **Table 4-22**.

Table 4-22 CloudTable link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	cloudtable_link
ZK Link	Obtain this parameter value from the cluster management page of CloudTable.	cloudtable-cdm-zk1.cloudtable.com:2181,cloudtable-cdm-zk2.cloudtable.com:2181

Parameter	Description	Example Value
IAM Authentication	If IAM authentication is enabled for the CloudTable cluster to be connected, set this parameter to **Yes**. Otherwise, set this to **No**. If you select **Yes**, enter the username, AK, and SK.	No
Username	Username used for accessing the CloudTable cluster	admin
AK	AK for accessing the CloudTable cluster. You need to create an access key for the current account and obtain an AK/SK pair.	-
SK	SK for accessing the CloudTable cluster. You need to create an access key for the current account and obtain an AK/SK pair.	-
Use Cluster Config	You can use the cluster configuration to simplify parameter settings for the Hadoop connection.	No
Cluster Config Name	This parameter is valid only when **Use Cluster Config** is set to **Yes**. Select a cluster configuration that has been created. For details about how to configure a cluster, see **Managing Cluster Configurations**.	hadoop_01

Click **Show Advanced Attributes**, and then click **Add** to add configuration attributes of other clients. The name and value of each attribute must be configured. You can click **Delete** to delete no longer used attributes.

4.18 Link to MongoDB

This link is used to transfer data from a third-party cloud MongoDB service or MongoDB created in the on-premises data center or ECS to a big data platform.

When connecting CDM to an on-premises MongoDB database, configure the parameters as described in **Table 4-23**.

Table 4-23 MongoDB link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	mongodb_link

Parameter	Description	Example Value
Server List	List of MongoDB server addresses. Enter each address in the format of *IP address or domain name of the database server*:*port number*, and separate the entered addresses with semicolons (;).	192.168.0.1:7300;192.168.0.2:7301
Database Name	Name of the MongoDB database to be connected	DB_mongodb
Username	Username for logging in to MongoDB	cdm
Password	Password for logging in to MongoDB	-
Direct Connection	This mode applies to the scenario where the network of the primary node is normal but the network of the replica node is abnormal. **NOTE** • Only one IP address can be configured for the server list in direct connection mode. • This mode applies to the scenario where the network of the primary node is normal but the network of the replica node is abnormal.	No
Link Attributes	Custom link attributes. The MongoDB attributes are supported. The unit is ms. The link attributes are as follows: • **socketTimeout**: The default value is **60000**. • **maxWaitTime**: The default value is **10000**. • **connectTimeout**. The default value is **10000**. • **serverSelectionTimeout**: The default value is **5000**.	socketTimeout=60000

4.19 Link to Cassandra

Table 4-24 Parameter description

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	mongodb_link

Parameter	Description	Example Value
Service node	An address of one node or addresses of multiple nodes. Separate addresses with semicolons (;). You are advised to configure multiple nodes at a time.	192.168.0.1;192.168.0.2
Port	Port number of the Cassandra node to be connected.	9042
Username	User name for connecting to Cassandra.	cdm
Password	Password for connecting to Cassandra.	-
Connection timeout duration	(Optional) Displayed when you click **Show Advanced Attributes**. Connection timeout interval, in seconds.	5
Read timeout duration	(Optional) Displayed when you click **Show Advanced Attributes**. Read timeout interval, in seconds. If the value is less than or equal to 0, no timeout occurs.	12

4.20 Link to DIS

When connecting CDM to DIS, configure the parameters as described in **Table 4-25**.

Table 4-25 DIS link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	dis_link
Region	Region where DIS is deployed	-
Endpoint	URL of DIS in the format of https://*Endpoint*. An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. You can obtain endpoints of the service from **Endpoints**.	-
AK	AK used for logging in to the DIS server. You need to create an access key for the current account and obtain an AK/SK pair.	-

Parameter	Description	Example Value
SK	SK used for logging in to the DIS server. You need to create an access key for the current account and obtain an AK/SK pair.	-
Project ID	Project ID of DIS	-

4.21 Link to Kafka

MRS Kafka

When connecting CDM to Kafka of MRS, configure the parameters as described in **Table 4-26**.

Table 4-26 MRS Kafka link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	kafka_link
Manager IP	Floating IP address of MRS Manager. Click **Select** next to the **Manager IP** text box to select an MRS cluster. CDM automatically fills in the authentication information.	127.0.0.1

Parameter	Description	Example Value
Username	Username used for logging in to MRS Manager To create a data connection for an MRS security cluster, do not use user **admin**. The **admin** user is the default management page user and cannot be used as the authentication user of the security cluster. You can create an MRS user and set **Username** and **Password** to the username and password of the created MRS user when creating an MRS data connection. **NOTE** • If the CDM cluster version is 2.9.0 or later and the MRS cluster version is 3.1.0 or later, the created user must have the permissions of the **Manager_viewer** role to create links on CDM. To perform operations on databases, tables, and columns of an MRS component, you also need to add the database, table, and column permissions of the MRS component to the user by following the instructions in the MRS documentation. • If the CDM cluster version is earlier than 2.9.0 or the MRS cluster version is earlier than 3.1.0, the created user must have the permissions of **Manager_administrator** or **System_administrator** to create links on CDM. • A user with only the **Manager_tenant** or **Manager_auditor** permission cannot create connections.	-
Password	Password used for logging in to MRS Manager	-
Authentication Method	Authentication method used for accessing MRS • **SIMPLE**: for non-security mode • **KERBEROS**: for security mode	Yes

Click **Show Advanced Attributes**, and then click **Add** to add configuration attributes of other clients. The name and value of each attribute must be configured. You can click **Delete** to delete no longer used attributes.

Apache Kafka

The Apache Kafka link is applicable to data migration of the third-party Hadoop in the local data center or ECS. You must use Direct Connect to connect to Hadoop in the local data center.

When connecting CDM to Kafka of Apache Hadoop, configure the parameters as described in **Table 4-27**.

Table 4-27 Parameter description

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	kafka_link
Kafka broker	IP address and port number of the Kafka broker	192.168.1.1:9092

Click **Show Advanced Attributes**, and then click **Add** to add configuration attributes of other clients. The name and value of each attribute must be configured. You can click **Delete** to delete no longer used attributes.

4.22 Link to DMS Kafka

When connecting CDM to DMS Kafka, configure the parameters as described in **Table 4-28**.

Table 4-28 DMS Kafka link parameter

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	dms_link
Service Type	DMS Kafka edition. Currently, only the Platinum edition is available.	Platinum
Kafka Broker	Address of a Kafka premium instance. The format is host:port.	-

Parameter	Description	Example Value
Kafka SASL_SSL	Whether to enable SSL authentication when a client connects to a Kafka premium instance. This function must be enabled if the SASL_SSL security protocol is enabled for the link to the DMS Kafka instance. If Kafka SASL_SSL is enabled, data will be encrypted before transmission for higher security, but performance will suffer. **NOTE** When SSL authentication is enabled, Kafka continuously parses the Kafka broker connection address as a domain name, which undermines performance. You are advised to add the self-mapping of the broker connection address to the **/etc/hosts** file on the ECS corresponding to the CDM cluster (search for the ECS based on the cluster IP address) so that the client can quickly resolve the broker of the instance. For example, if the Kafka broker address is 10.154.48.120, add the following self-mapping to the **/etc/hosts** file: 10.154.48.120 10.154.48.120	Yes
Username	Username for connecting to DMS Kafka. This parameter is displayed when **Kafka SASL_SSL** is enabled.	-
Password	Password for connecting to DMS Kafka. This parameter is displayed when **Kafka SASL_SSL** is enabled.	-
Kafka Properties	• If a security protocol is enabled for the link to the DMS Kafka instance, you must add a data encryption attribute, and set the attribute name to **security.protocol** and value to **SASL_SSL** or **SASL_PLAINTEXT** based on the security protocol of the Kafka instance. • If SASL authentication is enabled for the link to the DMS Kafka instance, you must add an authentication mode attribute, and set the attribute name to **sasl.mechanism** and value to **PLAIN** or **SCRAM-SHA-512** based on the SASL authentication mechanism configured for the Kafka instance (set the value to either **PLAIN** or **SCRAM-SHA-512** if both are supported).	-

4.23 Link to CSS

Huawei Cloud Cloud Search Service (CSS) is a fully hosted distributed search service powered by open-source Elasticsearch. CSS links can be used to migrate log files and database records to CSS for search and analysis using Elasticsearch.

NOTE

You are advised to use Logstash to import data to CSS. For details, see **Using Logstash to Import Data to Elasticsearch**.

Table 4-29 lists the parameters for a CSS link.

Table 4-29 CSS link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	css_link
Elasticsearch Server List	IP addresses or domain names (including the port numbers) of one or more Elasticsearch servers. The format is *ip:port*. Use semicolons (;) to separate multiple IP addresses.	192.168.0.1:9200;192.168.0.2:9200
Security Mode Authentication	Whether to enable security mode. If **Security Mode** has been enabled for the CSS cluster to be connected, set this parameter to **Yes**. Otherwise, set this to **No**.	Yes
Username	This parameter is displayed when **Security Mode Authentication** is set to **Yes**. It indicates the username used for connecting to CSS.	admin
Password	This parameter is displayed when **Security Mode Authentication** is set to **Yes**. It indicates the password used for connecting to CSS.	-
HTTPS Access	This parameter is displayed when **Security Mode Authentication** is set to **Yes**. This parameter specifies whether to enable HTTPS access. HTTPS access is more secure than HTTP access.	Yes

4.24 Link to Elasticsearch

Elasticsearch links can be used to connect to Elasticsearch services in third-party clouds and local data centers and on Elastic Cloud Servers (ECSs).

NOTE

The Elasticsearch connector only supports Elasticsearch clusters in non-security mode.

Table 4-30 lists the parameters for an Elasticsearch link.

Table 4-30 Elasticsearch link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	css_link
Elasticsearch Server List	IP addresses or domain names (including the port numbers) of one or more Elasticsearch servers. The format is *ip:port*. Use semicolons (;) to separate multiple IP addresses or domain names.	192.168.0.1:9200;192.168.0.2:9200

4.25 Link to a Dameng Database

When connecting CDM to a Dameng database, configure the parameters as described in Table 4-31.

Table 4-31 Parameters for a link to a Dameng database

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	dm_link
Database Server	IP address or domain name of the database to connect Click **Select** next to the text box and select a DWS or RDS DB instance in the displayed dialog box.	192.168.0.1
Port	Port of the database to connect	The port number varies depending on the database.

Parameter	Description	Example Value
Database Name	Name of the database to connect	dbname
Username	Username used for accessing the database This account must have the permissions required to read and write data tables and metadata.	cdm
Password	Password of the user	-
Driver Version	Select a driver version that adapts to the database type.	-
Fetch Size	(Optional) Displayed when you click **Show Advanced Attributes**. Number of rows obtained by each request. Set this parameter based on the data source and the job's data size. If the value is either too large or too small, the job may run for a long time.	1000
Link Attributes	(Optional) Click **Add** to add the JDBC connector attributes of multiple specified data sources. For details, see the JDBC connector document of the corresponding database.	sslmode=require
Reference Sign	(Optional) Delimiter between the names of the referenced tables or columns. For details, see the product documentation of the corresponding database.	'

4.26 Link to SAP HANA

Table 4-32 describes the SAP HANA link parameters.

Table 4-32 SAP HANA link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	sap_link
Database Server	IP address or domain name of the database to connect Click **Select** next to the text box to obtain the list of instances.	192.168.0.1

Parameter	Description	Example Value
Port	Port of the database to connect	The port number varies depending on the database.
Database Name	Name of the database to connect	dbname
Username	Username used for accessing the database. This user must have the permissions to read and write data tables and metadata.	cdm
Password	Password of the user	-
Use Agent	Whether to extract data from the data source through an agent	Yes
Agent	Click **Select** and select the agent created in **Managing Agents**.	-
Fetch Size	(Optional) Displayed when you click **Show Advanced Attributes**. Number of rows obtained by each request. Set this parameter based on the data source and the job's data size. If the value is either too large or too small, the job may run for a long time.	1000

Parameter	Description	Example Value
Link Attributes	(Optional) Click **Add** to add the JDBC connector attributes of multiple specified data sources. For details, see the JDBC connector document of the corresponding database. The following are some examples: • **connectTimeout=360000** and **socketTimeout=360000**: When a large amount of data needs to be migrated or the entire table is retrieved using query statements, the migration fails due to connection timeout. In this case, you can customize the connection timeout interval (ms) and socket timeout interval (ms) to prevent failures caused by timeout. • **useCursorFetch=false**: By default, **useCursorFetch** is enabled, indicating that the JDBC connector communicates with relational databases using a binary protocol. Some third-party systems may have compatibility issues, causing migration time conversion errors. In this case, you can disable this function. Open-source MySQL databases support the **useCursorFetch** parameter, and you do not need to set this parameter.	sslmode=require
Reference Sign	(Optional) Delimiter between the names of the referenced tables or columns. For details, see the product documentation of the corresponding database.	'

4.27 Link to a Database Shard

Sharding refers to the link to multiple backend data sources at the same time. The link can be used as the job source to migrate data from multiple data sources to other data sources. **Table 4-33** lists the link parameters.

Table 4-33 Database shard link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	my_link

Parameter	Description	Example Value
Username	Username used for accessing the database For a backend database A, this configuration takes effect only when no username and password are configured for A in the data source list. For a backend database B that has configured the username and password, this configuration does not takes effect.	cdm
Password	Password used for accessing the database. For a backend database A, this configuration takes effect only when no username and password are configured for A in the data source list. For a backend database B that has configured the username and password, this configuration does not takes effect.	-
Use Agent	Whether to extract data from the data source through an agent	Yes
Agent	Click **Select** and select the agent created in **Managing Agents**.	-
backendDatasource	Enter the type of the backend database. Currently, only MySQL is supported.	MySQL
Data Source List	Enter the IP address, port number, database name, account name, and password of the backend database, and separate them with colons (:). That is, ip:port:dbs:username:password. You can leave username:password empty. In this case, the username and password are used. If there are multiple backend databases, ensure that the table structures are the same and use vertical bars (|) to separate data sources. If the password contains a vertical bar (|) or colon (:), use a backslash (\) to escape the vertical bar. For example, **192.168.3.0:3306:cdm|192.168.2.2:3306:cdm:user:password** indicates that the IP address of the first backend database is **192.168.3.0**, the port number is **3306**, the database name is **cdm**, and the account name and password are configured in *user* and *password*. The IP address of the second backend database is **192.168.2.2**, the port number is **3306**, the database name is **cdm**, the account name is **user** and the password is **password**.	192.168.3.0:3306:cdm|192.168.2.2:3306:cdm:user:password

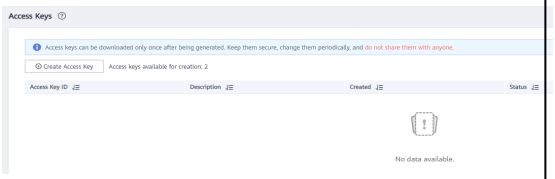
Parameter	Description	Example Value
Fetch Size	(Optional) Displayed when you click **Show Advanced Attributes**. Number of rows obtained by each request. Set this parameter based on the data source and the job's data size. If the value is either too large or too small, the job may run for a long time.	1000
Link Attributes	(Optional) Click **Add** to add the JDBC connector attributes of multiple specified data sources. For details, see the JDBC connector document of the corresponding database.	sslmode=require
Reference Sign	(Optional) Delimiter between the names of the referenced tables or columns. For details, see the product documentation of the corresponding database.	'

4.28 Link to MRS Hudi

Table 4-34 describes the MRS Hudi link parameters.

Table 4-34 Hudi link parameters

Parameter	Description	Example Value
Name	Link name	Hudilink
Manager IP	Floating IP address of MRS Manager. Click **Select** next to the **Manager IP** text box to select an MRS cluster. CDM automatically fills in the authentication information.	127.0.0.1
Authentication Method	Authentication method used for accessing MRS • **SIMPLE**: Select this for non-security mode. • **KERBEROS**: Select this for security mode.	KERBEROS
Account	Username for logging in to MRS Manager	cdm
Password	Password for logging in to MRS Manager	-
OBS storage support	Whether to support OBS storage. If the Hudi table data is stored in OBS, you need to enable this function.	Yes

Parameter	Description	Example Value
AK	This parameter is available when **OBS storage support** is set to **Yes**. AK and SK are used to log in to the OBS server. You need to create an access key for the current account and obtain an AK/SK pair. To obtain an access key, perform the following steps: 1. Log in to the management console, move the cursor to the username in the upper right corner, and select **My Credentials** from the drop-down list. 2. On the **My Credentials** page, choose **Access Keys**, and click **Create Access Key**. See **Figure 4-17**. **Figure 4-17** Clicking Create Access Key  3. Click **OK** and save the access key file as prompted. The access key file will be saved to your browser's configured download location. Open the **credentials.csv** file to view **Access Key Id** and **Secret Access Key**. **NOTE** • Only two access keys can be added for each user. • To ensure access key security, the access key is automatically downloaded only when it is generated for the first time and cannot be obtained from the management console later. Keep them properly.	-
SK		-

Parameter	Description	Example Value
OBS Test Path	This parameter is available when **OBS storage support** is set to **Yes**. Enter a complete file path. The permission to access the path will be verified through the metadata query API. **NOTE** • For object storage, the path must be accurate to object, for example, **obs://bucket/dir/test.txt**. Otherwise, a 404 error occurs. • For a parallel file system, the path must be accurate to directory, for example, **obs://bucket/dir**.	obs://bucket/dir/test.txt
Hive Properties	Names of the tables to be integrated. Use commas (,) to separate multiple table names. This parameter is mandatory and cannot contain spaces.	-

4.29 Link to MRS ClickHouse

Table 4-35 describes the MRS ClickHouse link parameters.

Table 4-35 ClickHouse link parameters

Parameter	Description	Example Value
Name	Link name	cklink
Database Server	IP address or domain name of the database to connect Log in to Manager of the cluster where the MRS ClickHouse data source is located, choose **Cluster** > **Services** > **ClickHouse** > **Instance**, and view the ClickHouseServer service IP address.	192.168.0.1

Parameter	Description	Example Value
Port	Port of the database to connect **NOTE** • If the Server node is used, enable **SSL Encryption** and set the default port. Log in to the Manager of the cluster where the MRS ClickHouse data source is located, choose **Cluster** > **Services** > **ClickHouse** > **Instance**, and set the default port of ClickHouseServer. For an MRS cluster in non-security mode, set it to the value of the **http_port** parameter. For an MRS cluster in security mode, set it to the value of the **https_port** parameter. • If the Balancer node is used, enable **SSL Encryption** and set the default port. Log in to the Manager of the cluster where the MRS ClickHouse data source is located, choose **Cluster** > **Services** > **ClickHouse** > **Instance**, and set the default port of ClickHouseBalancer. For an MRS cluster in non-security mode, set it to the value of the **lb_http_port** parameter. For an MRS cluster in security mode, set it to the value of the **lb_https_port** parameter. • If MRS ClickHouse is deployed in a security cluster, set this parameter to the default HTTPS port.	8123
Database Name	Name of the database to connect	dbname
Username	Username used for accessing the database. This user must have the permissions to read and write data tables and metadata.	cdm
Password	Password of the user	-
SSL Encryption	(Optional) If you set this parameter to **Yes**, CDM can connect to the database (on-premises databases excluded) in SSL encryption mode.	No
Reference Sign	(Optional) Delimiter between the names of the referenced tables or columns. For details, see the product documentation of the corresponding database.	'

4.30 Link to a ShenTong Database

Table 4-36 lists the parameters for a link to a ShenTong database.

Table 4-36 ShenTong database link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	st_link
Database Server	IP address or domain name of the database to connect Click **Select** next to the text box and select a ShenTong DB instance in the displayed dialog box.	192.168.0.1
Port	Port of the database to connect	3306
Database Name	Name of the database to connect	dbname
Username	Username used for accessing the database. This user must have the permissions to read and write data tables and metadata.	cdm
Password	Password of the user	-
Use Agent	Whether to extract data from the data source through an agent	Yes
Agent	Click **Select** and select the agent created in **Managing Agents**.	-
Reference Sign	(Optional) Delimiter between the names of the referenced tables or columns. For details, see the product documentation of the corresponding database.	'
Driver Version	Select a driver version that adapts to the database type.	-
Fetch Size	(Optional) Displayed when you click **Show Advanced Attributes**. Number of rows obtained by each request. Set this parameter based on the data source and the job's data size. If the value is either too large or too small, the job may run for a long time.	1000

Parameter	Description	Example Value
Link Attributes	(Optional) Click **Add** to add the JDBC connector attributes of multiple specified data sources. For details, see the JDBC connector document of the corresponding database. The following are some examples: • **connectTimeout=360000** and **socketTimeout=360000**: When a large amount of data needs to be migrated or the entire table is retrieved using query statements, the migration fails due to connection timeout. In this case, you can customize the connection timeout interval (ms) and socket timeout interval (ms) to prevent failures caused by timeout.	sslmode=require

5 Managing Jobs

5.1 Table/File Migration Jobs

Scenario

CDM supports table and file migration between homogeneous or heterogeneous data sources. For details about supported data sources, see **Supported Data Sources**.

Constraints

- The dirty data recording function depends on OBS.
- The JSON file of a job to be imported cannot exceed 1 MB.
- The size of a file to be transferred cannot exceed 1 TB.
- Field names of the source and destination parameters cannot contain ampersands (&) or number signs (%).

Prerequisites

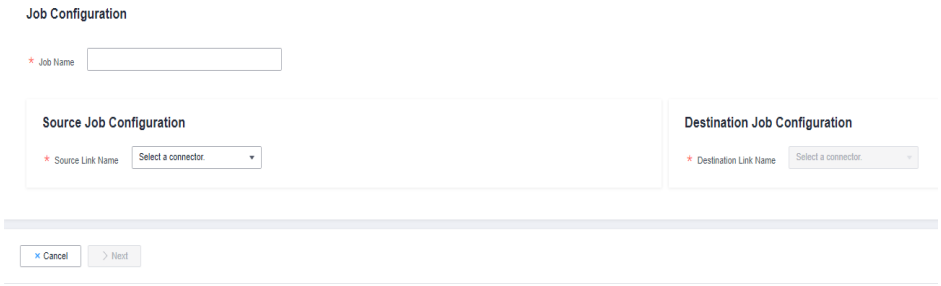
- A link has been created. For details, see **Creating a Link**.
- The CDM cluster can communicate with the data source.

Procedure

Step 1 Log in to the management console and choose **Service List** > **Cloud Data Migration**. In the left navigation pane, choose **Cluster Management**. Locate the target cluster and click **Job Management**.

Step 2 Choose **Table/File Migration** > **Create Job**. The page for configuring the job is displayed.

Figure 5-1 Creating a migration job

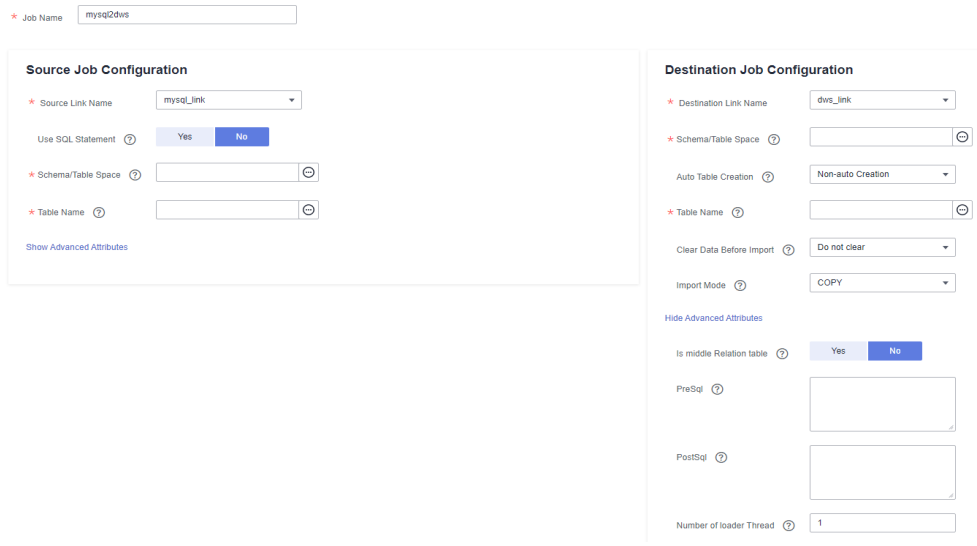


Step 3 Select the source and destination links.

- **Job Name**: Enter a string consisting of 1 to 240 characters. The name can contain digits, letters, hyphens (-), underscores (_), and periods (.), and cannot start with a hyphen (-) or period (.). An example value is **oracle2rds_t**.
- **Source Link Name**: Select the data source from which data will be exported.
- **Destination Link Name**: Select the data source to which data will be imported.

Step 4 Configure the source link parameters. **Figure 5-2** shows the job configurations for migrating MySQL to DWS.

Figure 5-2 Creating a job



The parameters vary with data sources. For details about the job parameters of other types of data sources, see **Table 5-1** and **Table 5-2**.

Table 5-1 Source link parameter description

Migration Source	Description	Parameter Settings
OBS	Data can be extracted in CSV, JSON, or binary format. Data extracted in binary format is free from file resolution, which ensures high performance and is more suitable for file migration.	For details, see **From OBS**.
• MRS HDFS • FusionInsight HDFS • Apache HDFS	HDFS data can be exported in CSV, Parquet, or binary format and can be compressed in multiple formats.	For details, see **From HDFS**.
• MRS HBase • FusionInsight HBase • Apache HBase • CloudTable Service	Data can be exported from MRS, FusionInsight HD, open source Apache Hadoop HBase, or CloudTable. You need to know all column families and field names of HBase tables.	For details, see **From HBase/CloudTable**.
• MRS Hive • FusionInsight Hive • Apache Hive	Data can be exported from Hive through the JDBC API. If the data source is Hive, CDM will automatically partition data using the Hive data partitioning file.	For details, see **From Hive**.
DLI	Data can be exported from DLI.	For details, see **From DLI**.
• FTP • SFTP	FTP and SFTP data can be exported in CSV, JSON, or binary format.	For details, see **From FTP/SFTP**.

Migration Source	Description	Parameter Settings
• HTTP	These connectors are used to read files with an HTTP/HTTPS URL, such as reading public files on the third-party object storage system and web disks. Currently, data can only be exported from the HTTP URLs.	For details, see **From HTTP**.
Data Warehouse Service	Data can be exported from DWS.	For details, see **From DWS**.
SAP HANA	Data can be exported from SAP HANA.	For details, see **From SAP HANA**.
• RDS for PostgreSQL • RDS for SQL Server • Microsoft SQL Server • PostgreSQL	Data can be exported from the cloud database services. The non-cloud databases can be those created in the on-premises data center or deployed on ECSs, or database services on the third-party clouds.	When data is exported from these data sources, CDM uses the JDBC API to extract data. The job parameters for the migration source are the same. For details, see **From PostgreSQL/SQL Server**.
MySQL	Data can be exported from a MySQL database.	For details, see **From MySQL**.
Oracle	Data can be exported from an Oracle database.	For details, see **From Oracle**.
Database Sharding	Data can be exported from a shard.	For details, see **From a Database Shard**.
• MongoDB • Document Database Service	Data can be exported from MongoDB or DDS.	For details, see **From MongoDB/DDS**.
Redis	Data can be exported from open source Redis.	For details, see **From Redis**.
Data Ingestion Service	Data can only be exported to Cloud Search Service (CSS).	For details, see **From DIS**.

Migration Source	Description	Parameter Settings
• Apache Kafka • DMS Kafka • MRS Kafka	Data can only be exported to Cloud Search Service (CSS).	For details, see **From Kafka/DMS Kafka**.
• Cloud Search Service • Elasticsearch	Data can be exported from CSS or Elasticsearch.	For details, see **From Elasticsearch or CSS**.
MRS Hudi	Data can be exported from MRS Hudi.	For details, see **From MRS Hudi**.
MRS ClickHouse	Data can be exported from MRS ClickHouse.	For details, see **From MRS ClickHouse**.
ShenTong database	Data can be exported from a ShenTong database.	For details, see **From a ShenTong Database**.
Dameng database	Data can be exported from a Dameng database.	For details, see **From a Dameng Database**.

Step 5 Configure job parameters for the migration destination based on **Table 5-2**.

Table 5-2 Parameter description

Migration Destination	Description	Parameter Settings
OBS	Files (even in a large volume) can be batch migrated to OBS in CSV or binary format.	For details, see **To OBS**.
MRS HDFS	You can select a compression format when importing data to HDFS.	For details, see **To HDFS**.
MRS HBase CloudTable Service	Data can be imported to HBase. The compression algorithm can be set when a new HBase table is created.	For details, see **To HBase/ CloudTable**.
MRS Hive	Data can be rapidly imported to MRS Hive.	For details, see **To Hive**.

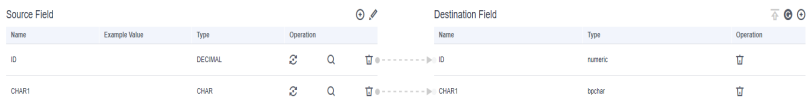
Migration Destination	Description	Parameter Settings
• MySQL • SQL Server • PostgreSQL	Data can be imported to cloud database services.	For details about how to use the JDBC API to import data, see **To MySQL/SQL Server/ PostgreSQL**.
DWS	Data can be imported to DWS.	For details, see **To DWS**.
Oracle	Data can be imported to an Oracle database.	For details, see **To Oracle**.
DLI	Data can be imported to DLI.	For details, see **To DLI**.
Elasticsearchor Cloud Search Service (CSS)	Data can be imported to CSS.	For details, see **To Elasticsearch/CSS**.
MRS Hudi	Data can be rapidly imported to MRS Hudi.	For details, see **To MRS Hudi**.
MRS ClickHouse	Data can be rapidly imported to MRS ClickHouse.	For details, see **To MRS ClickHouse**.
MongoDB	Data can be rapidly imported to MongoDB.	For details, see **To MongoDB**.

Step 6 After the parameters are configured, click **Next**. The **Map Field** tab page is displayed.

If files are migrated between FTP, SFTP, OBS, and HDFS and the migration source's **File Format** is set to **Binary**, files will be directly transferred, free from field mapping.

In other scenarios, CDM automatically maps fields of the source table and the destination table. You need to check whether the mapping and time format are correct. For example, check whether the source field type can be converted into the destination field type.

Figure 5-3 Field mapping

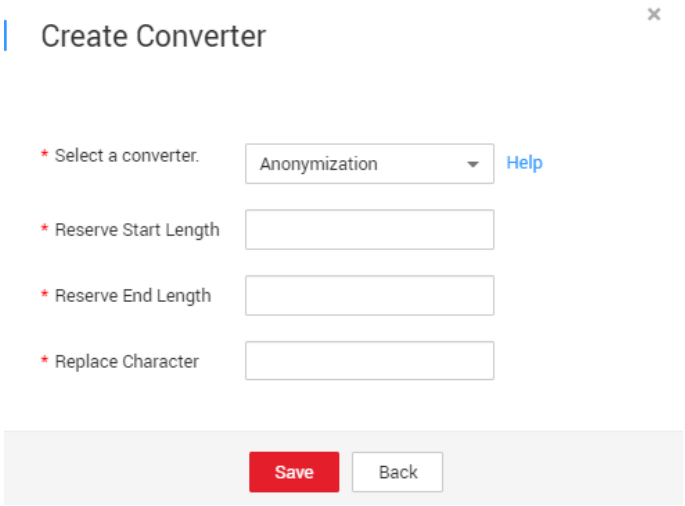


NOTE

- If the fields from the source and destination do not match, you can drag the fields to make adjustments.
- On the **Map Field** tab page, if CDM fails to obtain all columns by obtaining sample values (for example, when data is exported from HBase, CloudTable, or MongoDB, or when data is migrated from SFTP/FTP to DLI, there is a high probability that CDM failed to obtain all columns), you can click and select **Add a new field** to add new fields to ensure that the data imported to the migration destination is complete.
- When a relational database, Hive, DLI, or MRS Hudi is used as the migration source, sample values cannot be obtained.
- On the **Map Field** page, you can click to add custom constants, variables, and expressions.
- Column names are displayed when the source of the migration job is OBS, CSV files are to be migrated, and parameter **Extract first row as columns** is set to **Yes**.
- When SQLServer is the destination, fields of the timestamp type cannot be written. You must change their type (for example, to datatime) so that they can be written.
- When Hive serves as the source, data of the array and map types can be read.
- Field mapping is not involved when the binary format is used to migrate files to files.
- If the data is imported to DWS, you need to select the distribution columns in the destination fields. You are advised to select the distribution columns according to the following principles:
  1. Use the primary key as the distribution column.
  2. If multiple data segments are combined as primary keys, specify all primary keys as the distribution column.
  3. In the scenario where no primary key is available, if no distribution column is selected, DWS uses the first column as the distribution column by default. As a result, data skew risks exist.

Step 7 CDM supports field conversion. Click and then click **Create Converter**.

Figure 5-4 Creating a converter



CDM supports the following converters:

- **Anonymization**: hides key data in the character string.

For example, if you want to convert **12345678910** to **123****8910**, configure the parameters as follows:

- Set **Reserve Start Length** to **3**.
- Set **Reserve End Length** to **4**.
- Set **Replace Character** to *****.

- **Trim** automatically deletes the spaces before and after the character string.
- **Reverse string** automatically reverses a character string. For example, reverse **ABC** into **CBA**.
- **Replace string** replaces the specified character string.
- **Expression conversion** uses the JSP expression language (EL) to convert the current field or a row of data. For details, see **Field Conversion**.
- **Remove line break** deletes the newline characters, such as \n, \r, and \r\n from the field.

NOTE

If **Use SQL Statement** is set to **Yes** in the source job configuration, converters cannot be created.

Step 8 Click **Next**, set job parameters, and click **Show Advanced Attributes** to display and configure optional parameters.

Figure 5-5 Task parameters

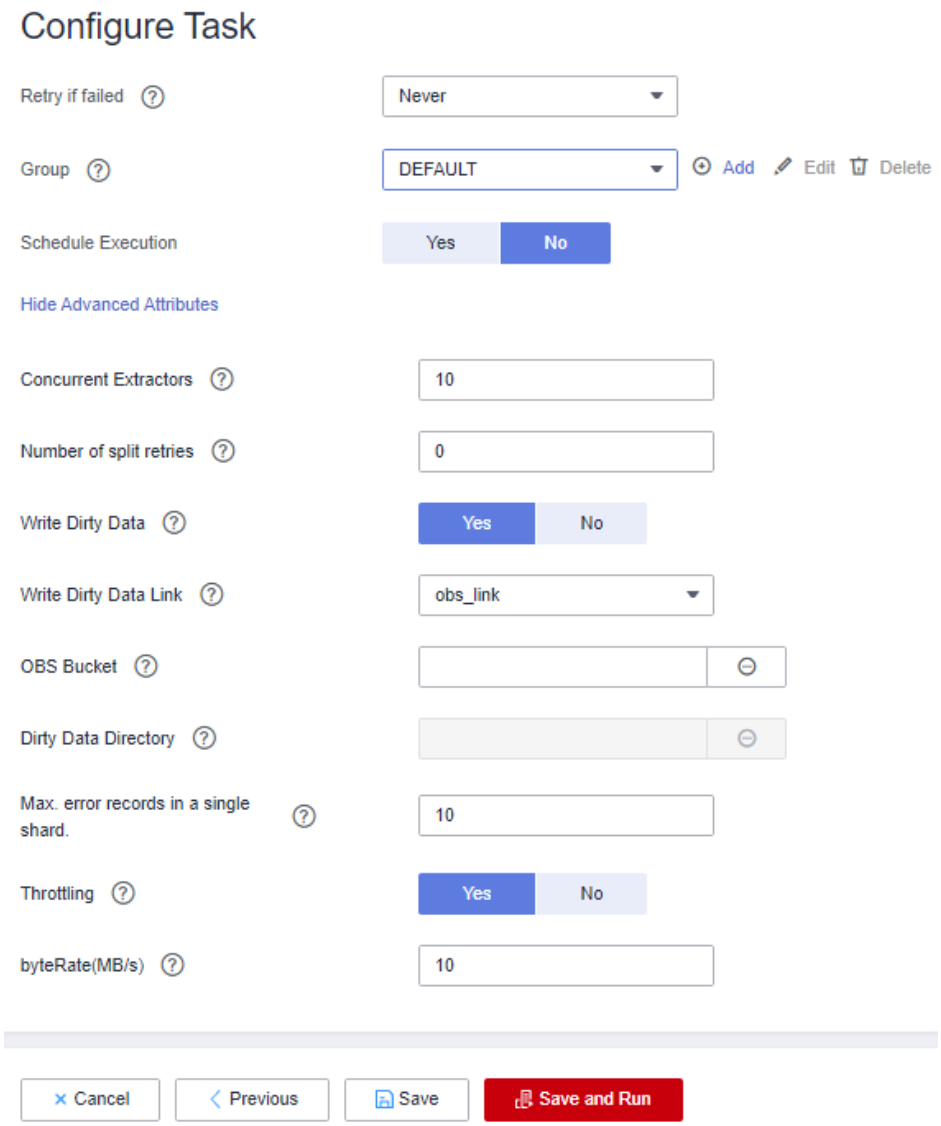


Table 5-3 describes related parameters.

Table 5-3 Parameter description

Parameter	Description	Example Value
Retry upon Failure	You can select **Retry 3 times** or **Never**. You are advised to configure automatic retry for only file migration jobs or database migration jobs with **Import to Staging Table** enabled to avoid data inconsistency caused by repeated data writes. **NOTE** If you want to set parameters in DataArts Studio DataArts Factory to schedule the CDM migration job, do not configure this parameter. Instead, set parameter **Retry upon Failure** for the CDM node in DataArts Factory.	Never
Job	Select a group where the job resides. The default group is **DEFAULT**. On the **Job Management** page, jobs can be displayed, started, or exported by group.	DEFAULT
Schedule Execution	If you select **Yes**, you can set the start time, cycle, and validity period of a job. For details, see **Scheduling Job Execution**. **NOTE** If you use DataArts Studio DataArts Factory to schedule the CDM migration job and configure this parameter, both configurations take effect. To ensure unified service logic and avoid scheduling conflicts, enable job scheduling in DataArts Factory and do not configure a scheduled task for the job in DataArts Migration.	No

Parameter	Description	Example Value
Concurrent Extractors	Configure the number of tasks to be split from a CDM job. CDM migrates data through data migration jobs. It works in the following way: 1. When data migration jobs are submitted, CDM splits each job into multiple tasks based on the **Concurrent Extractors** parameter in the job configuration. **NOTE** Jobs for different data sources may be split based on different dimensions. Some jobs may not be split based on the **Concurrent Extractors** parameter. 2. CDM submits the tasks to the running pool in sequence. Tasks (defined by **Maximum Concurrent Extractors**) run concurrently. Excess tasks are queued. By setting appropriate values for this parameter and the **Maximum Concurrent Extractors** parameter, you can accelerate migration. Configure the number of concurrent extractors based on the following rules: 1. When data is to be migrated to files, CDM does not support multiple concurrent tasks. In this case, set a single process to extract data. 2. If each row of the table contains less than or equal to 1 MB data, data can be extracted concurrently. If each row contains more than 1 MB data, it is recommended	1

Parameter	Description	Example Value
	that data be extracted in a single thread. 3. Set **Concurrent Extractors** for a job based on **Maximum Concurrent Extractors** for the cluster. It is recommended that **Concurrent Extractors** is less than **Maximum Concurrent Extractors**. 4. If the destination is DLI, you are advised to set the number of concurrent extractors to 1. Otherwise, data may fail to be written. The maximum number of concurrent extractors for a cluster varies depending on the CDM cluster flavor. You are advised to set the maximum number of concurrent extractors to twice the number of vCPUs of the CDM cluster. For example, the maximum number of concurrent extractors for a cluster with 8 vCPUs and 16 GB memory is 16.	
Concurrent Loaders	Number of Loaders to be concurrently executed This parameter is displayed only when HBase or Hive serves as the destination data source.	3
Number of split retries	Number of retries when a split fails to be executed. Value **0** indicates that no retry will be performed.	0

Parameter	Description	Example Value
Write Dirty Data	Whether to record dirty data. By default, this parameter is set to **No**. Dirty data in CDM refers to the data in invalid format. If the source data contains dirty data, you are advised to enable this function. Otherwise, the migration job may fail. **NOTE** Dirty data can only be written to OBS paths. Therefore, this parameter is available only when an OBS link is available.	Yes
Write Dirty Data Link	This parameter is displayed only when **Write Dirty Data** is set to **Yes**. You can only select an OBS link.	obs_link
OBS Bucket	This parameter is displayed only when **Write Dirty Data Link** is a link to OBS. Name of the OBS bucket to which the dirty data will be written.	dirtydata
Dirty Data Directory	This parameter is displayed only when **Write Dirty Data** is set to **Yes**. Dirty data is stored in the directory for storing dirty data on OBS. Dirty data is saved only when this parameter is configured. You can go to this directory to query data that fails to be processed or is filtered out during job execution, and check the source data that does not meet conversion or cleaning rules.	/user/dirtydir

Parameter	Description	Example Value
Max. Error Records in a Single Shard	This parameter is displayed only when **Write Dirty Data** is set to **Yes**. When the number of error records of a single map exceeds the upper limit, the job will automatically terminate and the imported data cannot be rolled back. You are advised to use a temporary table as the destination table. After the data is imported, rename the table or combine it into the final data table.	0
Throttling	Enabling throttling reduces the read pressure on the source. It controls the CDM transmission rate, not the NIC traffic. **NOTE** • Throttling can be enabled for non-binary file migration jobs. • To configure throttling for multiple jobs, multiply the rate by the number of concurrent jobs. • Throttling is not supported for binary transmission between files.	Yes
Max. error records in a single shard	Maximum rate for a job. To configure throttling for multiple jobs, multiply the rate by the number of concurrent jobs. **NOTE** The rate is an integer greater than 1.	20

Step 9 Click **Save** or **Save and Run**. On the page displayed, you can view the job status.

NOTE

The job status can be **New**, **Pending**, **Booting**, **Running**, **Failed**, **Succeeded**, or **Stopped**.

Pending indicates that the job is waiting to be scheduled by the system, and **Booting** indicates that the data to be migrated is being analyzed.

----End

5.2 Creating an Entire Database Migration Job

Scenario

CDM supports entire DB migration between homogeneous and heterogeneous data sources. The migration principles are the same as those in **Table/File Migration Jobs**. Each type of Elasticsearch, each key prefix of Redis, or each collection of MongoDB can be executed concurrently as a subtask.

NOTE

Each time an entire DB migration job is executed, its subtasks are recreated based on the configuration of the migration job. You cannot modify the subtasks and then run the migration job again.

Supported Data Sources lists the data sources supporting entire database migration.

Constraints

Field names of the source and destination parameters cannot contain ampersands (&) or number signs (%).

Prerequisites

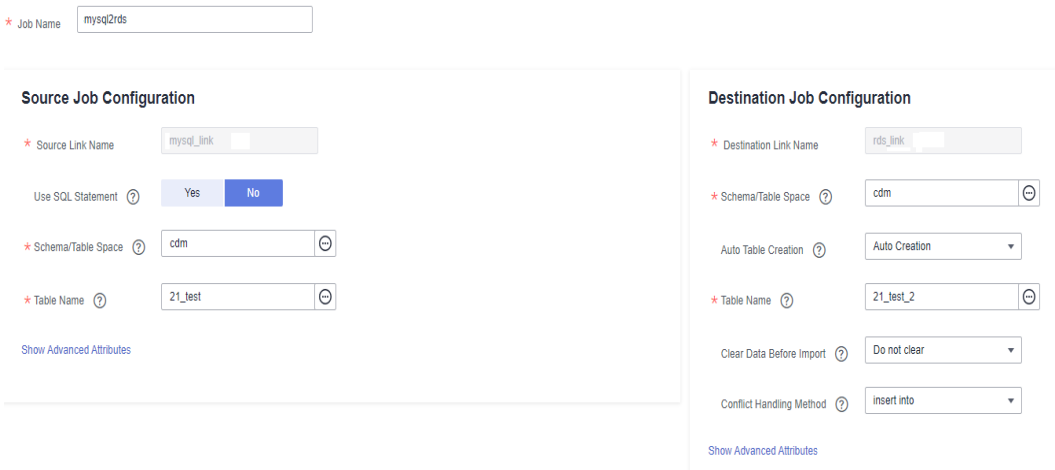
- A link has been created. For details, see **Creating a Link**.
- The CDM cluster can communicate with the data source.

Procedure

Step 1 Log in to the management console and choose **Service List** > **Cloud Data Migration**. In the left navigation pane, choose **Cluster Management**. Locate the target cluster and click **Job Management**.

Step 2 Choose **Entire DB Migration** > **Create Job**. The page for configuring the job is displayed.

Figure 5-6 Creating an entire DB migration job



Step 3 Configure the related parameters of the source database according to **Table 5-4**.

Table 5-4 Parameter description

Source Database	Parameter	Description	Example Value
• DWS • MySQL • PostgreSQL • SQL Server • Oracle • SAP HANA	Schema/ Tablespace	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. If the desired schema or tablespace is not displayed, check whether the login account has the permissions required to query metadata.	schema
	WHERE Clause	WHERE clause used to specify the tables to be extracted. This parameter applies to all subtables in the entire DB migration. If this parameter is not set, the entire table is extracted. If the table to be migrated does not contain the fields specified by the WHERE clause, the migration will fail. You can set a date macro variable to extract data generated on a specific date. For details, see **Incremental Migration of Relational Databases**.	age > 18 and age <= 60
	Null in Partition Column	Whether a partition field can be null	Yes

Source Database	Parameter	Description	Example Value
Hive	Database Name	Name of the database to be migrated. The user configured in the source link must have the permission to read the database.	hivedb
HBase CloudTable	Start Time	Start time (included). The format is *yyyy-MM-dd hh:mm:ss*. The dateformat time macro variable function is supported. Examples: **2017-12-31 20:00:00**, **${dateformat(yyyy-MM-dd, -1, DAY)} 02:00:00**, and **${dateformat(yyyy-MM-dd HH:mm:ss, -1, DAY)}**	-
	End Time	End time (excluded) The format is *yyyy-MM-dd hh:mm:ss*. The dateformat time macro variable function is supported. Examples: **2018-01-01 20:00:00**, **${dateformat(yyyy-MM-dd, -1, DAY)} 02:00:00**, and **${dateformat(yyyy-MM-dd HH:mm:ss, -1, DAY)}**	-
Redis	Key Filter Character	Filter character used to determine the keys to be migrated For example, if the value of this parameter is **a***, all asterisks (*) will be migrated.	-
DDS	Database Name	Name of the database from which data is to be migrated. The user configured in the source link must have the permission to read the database.	ddsdb

Source Database	Parameter	Description	Example Value
	Query Filter	Filter used to match documents. Example: **{HTTPStatusCode: {$gt:"400",$lt:"500"}, HTTPMethod:"GET"}**	-

Step 4 Configure the related parameters, from **Table 5-5**, for the destination cloud service.

Table 5-5 Destination job parameters

Destination Database	Parameter	Description	Example Value
• RDS for MySQL • RDS for PostgreSQL • RDS for SQL Server	-	For details about the destination job parameters required for entire DB migration to an RDS database, see **To MySQL/SQL Server/ PostgreSQL**.	schema
DWS	-	For details about the destination job parameters required for entire DB migration to DWS, see **To DWS**.	-
MRS Hive	-	For details about the destination job parameters required for entire DB migration to MRS HIVE, see **To Hive**.	hivedb
MRS HBase CloudTable	-	For details about the destination job parameters required for entire DB migration to MRS HBase or CloudTable, see **To HBase/ CloudTable**.	Yes
Redis	Clear Database	Clears the database data before data import.	Yes
DDS	Database Name	Name of the database from which data is to be migrated. The user configured in the source link must have the permission to read the database.	mongodb
	Migration Behavior	Select **Add** or **Replace**.	-

Step 5 If a relational database is migrated, after job parameters are configured, click **Next** to access the page for selecting tables. You can select the tables to be migrated to the migration destination based on your requirements.

Step 6 Click **Next** and set job parameters.

Figure 5-7 Task parameters

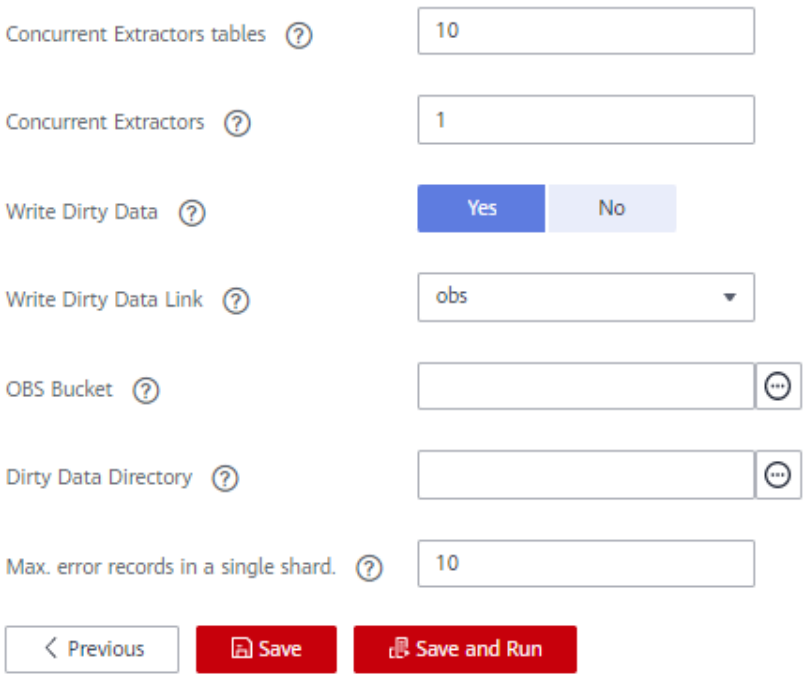


Table 5-6 describes related parameters.

Table 5-6 Task configuration parameters

Parameter	Description	Example Value
Concurrent Tables	Number of tables to be concurrently executed	3
Concurrent Extractors	Number of extractors to be concurrently executed. Generally, retain the default value.	1
Write Dirty Data	Whether to record dirty data. By default, this parameter is set to **No**.	Yes
Write Dirty Data Link	This parameter is only displayed when **Write Dirty Data** is set to **Yes**. Only links to OBS support dirty data writes.	obs_link
OBS Bucket	This parameter is only displayed when **Write Dirty Data Link** is a link to OBS. Name of the OBS bucket to which the dirty data will be written.	dirtydata

Parameter	Description	Example Value
Dirty Data Directory	This parameter is only displayed when **Write Dirty Data** is set to **Yes**. Directory for storing dirty data on OBS. Dirty data is saved only when this parameter is configured. You can go to this directory to query data that fails to be processed or is filtered out during job execution, and check the source data that does not meet conversion or cleaning rules.	/user/dirtydir
Max. Error Records in a Single Shard	This parameter is only displayed when **Write Dirty Data** is set to **Yes**. When the number of error records of a single map exceeds the upper limit, the job will automatically terminate and the imported data cannot be rolled back. You are advised to use a temporary table as the destination table. After the data is imported, rename the table or combine it into the final data table.	0

Step 7 Click **Save** or **Save and Run**.

When the job starts running, a sub-job will be generated for each table. You can click the job name to view the sub-job list.

----End

NOTE

During the migration of an entire Oracle database to Hudi, if you select a view or a table that has no primary key at the source, automatic table creation is not supported.

5.3 Source Job Parameters

5.3.1 From OBS

If the source link of a job is an **OBS link**, configure the source job parameters based on **Table 5-7**.

Advanced attributes are optional and not displayed by default. You can click **Show Advanced Attributes** to display them.

Table 5-7 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Bucket Name	Name of the bucket from which data will be migrated	BUCKET_2
	Source Directory/File	This parameter is available only when **Pull List File** is set to **No**. Directory or file path from which data will be extracted. You can enter a maximum of 50 file paths. By default, the file paths are separated by vertical bars (|). You can also customize a file separator. For details, see **Migration of a List of Files**. Directory from which data is to be migrated. All files (including all nested subdirectories and their subfiles) in the directory will be migrated. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	FROM/example.csv
	File Format	Format in which CDM parses data. The options are as follows: • **CSV**: Source files will be migrated to tables after being converted to CSV format. • **Binary**: Files (even not in binary format) will be transferred directly. It is used for file copy. • **JSON**: Source files will be migrated to tables after being converted to JSON format.	CSV

Category	Parameter	Description	Example Value
	Pull List File	This parameter is displayed only when **File Format** is set to **Binary**. If the pull list file function is enabled, the content of a file (such as a .txt file) in an OBS bucket can be read as the list of files to be migrated. The content in the file must be the absolute path of the file to be migrated (rather than a directory). For example, the content is as follows: /052101/DAY20211110.data /052101/DAY20211111.data	Yes
	OBS Link of List File	This parameter is available only when **Pull List File** is set to **Yes**. You can select the OBS link where the list file is located.	OBS_test_link
	OBS Bucket of entries files	This parameter is available only when **Pull List File** is set to **Yes**. It indicates the name of the OBS bucket where the list file is located.	01
	Path/ Directory of entries files	This parameter is available only when **Pull List File** is set to **Yes**. It indicates the absolute path or directory of the list file in the OBS bucket. You are advised to select the absolute path of the file. If you select a directory, files in subdirectories can also be migrated. However, if the number of files in the directory is too large, the cluster memory may become insufficient.	/0521/ Lists.txt
	JSON Type	This parameter is displayed only when **File Format** is set to **JSON**. Type of a JSON object stored in a JSON file. The options are **JSON object** and **JSON array**.	JSON object
	JSON Reference Node	This parameter is used only when **File Format** is set to **JSON** and **JSON Type** is set to **JSON Object**. CDM parses the data under the JSON node. If the node's corresponding data is a JSON array, the system will extract data from the array in the same pattern. Use periods (.) to separate multi-layer nested JSON nodes.	data.list

Category	Parameter	Description	Example Value
Advanced attributes	Line Separator	Lind feed character in a file. By default, the system automatically identifies **\n**, **\r**, and **\r\n**. This parameter is displayed only when **File Format** is set to **CSV**.	\n
	Field Delimiter	Character used to separate fields in the file. To set the **Tab** key as the delimiter, set this parameter to **\t**. This parameter is displayed only when **File Format** is set to **CSV**.	,
	Use Quote Character	If you set this parameter to **Yes**, the field delimiters in the encircling symbol are regarded as a part of the string value. Currently, the default encircling symbol of CDM is **"**.	No
	Using Escape Char	If you select **Yes**, the backslash (\) in the data row is used as an escape character. If you select **No**, the backslash (\) in the CSV file will not be escaped. CSV supports only the backslash (\) as the escape character.	Yes
	Use RE to Separate Fields	Whether to use regular expressions to separate fields. If you set this parameter to **Yes**, **Field Delimiter** becomes invalid. This parameter is displayed only when **File Format** is set to **CSV**.	Yes
	Regular Expression	Regular expression used to separate fields. For details about regular expressions, see **Regular Expressions for Separating Semi-structured Text**.	^(\d.*\d) (\w*) \[(.*) \] ([\w\.]*) (\w.*).*
	Use First N Rows as Header	This parameter is displayed only when **File Format** is set to **CSV**. When you migrate a CSV file to a table, CDM writes all data to the table by default. If you set this parameter to **Yes**, CDM uses the first N rows of the CSV file as the heading row and does not write the row to the destination table.	No

Category	Parameter	Description	Example Value
	The Number of Header Rows	This parameter is available when **Use First N Rows as Header** is set to **Yes**. It specifies the number of header rows to be skipped during data extraction. **NOTE** The number of header rows cannot be empty. The value is an integer from 1 to 99.	1
	Extract first row as columns	This parameter is available when **Use First N Rows as Header** is set to **Yes**. It specifies whether to parse the first row of the header as a column name. The column name is displayed in the source field during field mapping configuration. **NOTE** • If the number of header rows is greater than 1, only the first row of the header can be parsed as the column name. • The column name cannot contain the ampersand (&). Otherwise, the job migration fails. If the column name contains the ampersand (&), you must change it in the CSV file to ensure successful migration.	Yes
	Encoding Type	Encoding type, for example, **UTF-8** or **GBK**. You can set the encoding type for text files only. This parameter is invalid when **File Format** is set to **Binary**.	GBK
	Compression Format	The options are as follows: • **NONE**: Files in all formats can be transferred. • **GZIP**: Only files in gzip format can be transferred. • **ZIP**: Only files in Zip format can be transferred. • **TAR.GZ**: Files in TAR.GZ format are transferred.	NONE

Category	Parameter	Description	Example Value
	Compressed File Suffix	This parameter is displayed when **Compression Format** is not **NONE**. This parameter specifies the extension of the files to be decompressed. The decompression operation is performed only when the file name extension is used in a batch of files. Otherwise, files are transferred in the original format. If you enter ***** or leave the parameter blank, all files are decompressed.	*
	Start Job by Marker File	Whether to start a job by a marker file. A job is only started if there is a marker file for starting the job in the source path. If there is no marker file, the job will be suspended for a period of time specified by **Suspension Period**.	No
	Marker File	Name of the marker file for starting a job. If you specify a marker file, the migration job is executed only when the marker file exists in the source path. The marker file will not be migrated.	ok.txt
	Suspension Period	Waiting period for a marker file. If you set **Start Job by Marker File** to **Yes** but there is no marker file in the source path, the job fails when the suspension period times out. If you set this parameter to **0** and there is no marker file in the source path, the job will fail immediately. Unit: second	10
	File Separator	File separator. If you enter multiple file paths in **Source Directory/Files**, CDM uses the file separator to identify files. The default value is **|**.	|
	Filter Type	Only paths or files that meet the filtering conditions are transferred. The options are **None**, **Wildcard**, and **Regex**. For details, see **Incremental File Migration**.	Wildcard

Category	Parameter	Description	Example Value
	Directory Filter	If you set **Filter Type** to **Wildcard** or **Regex**, enter a wildcard character to filter paths. The paths that meet the filtering condition are migrated. You can configure multiple paths separated by commas (,). **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	*input
	File Filter	If you set **Filter Type** to **Wildcard** or **Regex**, you can enter a wildcard character to search for files in a specified path. The files that meet the search criteria are migrated. You can configure multiple files separated by commas (,). **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	*.csv,*.txt
	Time Filter	If you select **Yes**, files are transferred based on their modification time.	Yes

Category	Parameter	Description	Example Value
	Minimum Timestamp	If you set **Filter Type** to **Time Filter**, and specify a point in time for this parameter, only the files modified at or after the specified time are transferred. The time format must be *yyyy-MM-dd HH:mm:ss*. This parameter can be set to a macro variable of date and time. For example, **${timestamp(dateformat(yyyy-MM-dd HH:mm:ss,-90,DAY))}** indicates that only files generated within the latest 90 days are migrated. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	2019-06-01 00:00:00
	Maximum Timestamp	If you set **Filter Type** to **Time Filter**, and specify a point in time for this parameter, only the files modified before the specified time are transferred. The time format must be *yyyy-MM-dd HH:mm:ss*. This parameter can be set to a macro variable of date and time. For example, **${timestamp(dateformat(yyyy-MM-dd HH:mm:ss))}** indicates that only the files whose modification time is earlier than the current time are migrated. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	2019-07-01 00:00:00
	Disregard Non-existent Path or File	If this is set to **Yes**, the job can be successfully executed even if the source path does not exist.	No

Category	Parameter	Description	Example Value
	MD5 File Extension	This parameter is displayed only when **File Format** is set to **Binary**. This parameter is used to check whether the files extracted by CDM are consistent with source files. For details, see **MD5 Verification**.	.md5

NOTE

1. CDM supports incremental file migration (by skipping repeated files), but does not support resumable transfer.

   For example, if three files are to be migrated and the second file fails to be migrated due to the network fault. When the migration task is started again, the first file is skipped. The second file, however, cannot be migrated from the point where the fault occurs, but can only be migrated again.
2. During file migration, a single task supports millions of files. If there are too many files in the directory to be migrated, you are advised to split the files into different directories and create multiple tasks.

5.3.2 From HDFS

If the source link of a job is an **HDFS link**, that is, if data is exported from MRS HDFS, FusionInsight HDFS, or Apache HDFS, configure the source job parameters based on **Table 5-8**.

Table 5-8 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Source Link Name	Select a type from the drop-down list box.	hdfs_to_cdm

Category	Parameter	Description	Example Value
	Source Directory/ File	This parameter is available only when **Pull List File** is set to **No**. Directory or file path from which data will be extracted. Directory from which data is to be migrated. All files (including all nested subdirectories and their subfiles) in the directory will be migrated. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	/user/cdm/
	File Format	File format used when transferring data. The options are as follows: • **CSV**: Source files will be migrated to tables after being converted to CSV format. • **Binary**: Files (even not in binary format) will be transferred directly. It is used for file copy. • **Parquet**: Source files will be migrated to tables after being converted to Parquet format.	CSV

Category	Parameter	Description	Example Value
	Pull List File	This parameter is displayed only when **File Format** is set to **Binary**. If the pull list file function is enabled, the content of a file (such as a .txt file) in an OBS bucket can be read as the list of files to be migrated. The content in the file must be the absolute path of the file to be migrated (rather than a directory). The following is example content: /mrs/job-properties/application_1634891604621_0014/job.properties /mrs/job-properties/application_1634891604621_0029/job.properties	Yes
	OBS Link of List File	This parameter is available only when **Pull List File** is set to **Yes**. You can select the OBS link where the list file is located.	OBS_test_link
	OBS Bucket of entries files	This parameter is available only when **Pull List File** is set to **Yes**. It indicates the name of the OBS bucket where the list file is located.	01
	Path/Directory of entries files	This parameter is available only when **Pull List File** is set to **Yes**. It indicates the absolute path or directory of the list file in the OBS bucket.	/0521/Lists.txt
Advanced attributes	Line Separator	Lind feed character in a file. By default, the system automatically identifies **\n**, **\r**, and **\r\n**. This parameter is displayed only when **File Format** is set to **CSV**.	\n
	Field Delimiter	Character used to separate fields in the file. To set the **Tab** key as the delimiter, set this parameter to **\t**. This parameter is displayed only when **File Format** is set to **CSV**.	,

Category	Parameter	Description	Example Value
	Use First Row as Header	This parameter is displayed only when **File Format** is set to **CSV**. When you migrate a CSV file to a table, CDM writes all data to the table by default. If you set this parameter to **Yes**, CDM uses the first N rows of the CSV file as the heading row and does not write the row to the destination table.	No
	Encoding Type	Encoding type, for example, **UTF-8** or **GBK**. You can set the encoding type for text files only. This parameter is invalid when **File Format** is set to **Binary**.	GBK
	Start Job by Marker File	Whether to start a job by a marker file. A job is only started if there is a marker file for starting the job in the source path. If there is no marker file, the job will be suspended for a period of time specified by **Suspension Period**.	ok.txt
	Filter Type	Only paths or files that meet the filtering conditions are transferred. The options are **None**, **Wildcard**, and **Regex**. For details, see **Incremental File Migration**.	-

Category	Parameter	Description	Example Value
	Directory Filter	If you set **Filter Type** to **Wildcard** or **Regex**, enter a wildcard character to filter paths. The paths that meet the filtering condition are migrated. You can configure multiple paths separated by commas (,). **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	*input
	File Filter	If you set **Filter Type** to **Wildcard** or **Regex**, you can enter a wildcard character to search for files in a specified path. The files that meet the search criteria are migrated. You can configure multiple files separated by commas (,). **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	*.csv
	Time Filter	If you select **Yes**, files are transferred based on their modification time.	Yes

Category	Parameter	Description	Example Value
	Minimum Timestamp	If you set **Filter Type** to **Time Filter**, and specify a point in time for this parameter, only the files modified at or after the specified time are transferred. The time format must be *yyyy-MM-dd HH:mm:ss*. This parameter can be set to a macro variable of date and time. For example, **${timestamp(dateformat(yyyy-MM-dd HH:mm:ss,-90,DAY))}** indicates that only files generated within the latest 90 days are migrated. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	2019-07-01 00:00:00

Category	Parameter	Description	Example Value
	Maximum Timestamp	If you set **Filter Type** to **Time Filter**, and specify a point in time for this parameter, only the files modified before the specified time are transferred. The time format must be *yyyy-MM-dd HH:mm:ss*. This parameter can be set to a macro variable of date and time. For example, **${timestamp(dateformat(yyyy-MM-dd HH:mm:ss))}** indicates that only the files whose modification time is earlier than the current time are migrated. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	2019-07-30 00:00:00
	Create Snapshot	If you set this parameter to **Yes**, CDM creates a snapshot for the source directory to be migrated (the snapshot cannot be created for a single file) before it reads files from HDFS. Then CDM migrates the data in the snapshot. Only the HDFS administrator can create a snapshot. After the CDM job is completed, the snapshot is deleted.	No

Category	Parameter	Description	Example Value
	Encryption	This parameter is displayed only when **File Format** is set to **Binary**. If the source data is encrypted, CDM can decrypt the data before exporting it. Select whether to decrypt the source data and select a decryption algorithm. The options are as follows: • **NONE**: Export data without decrypting it. • **AES-256-GCM**: The AES 256-bit encryption algorithm is used to encrypt data. Currently, only the AES-256-GCM (NoPadding) encryption algorithm is supported. This parameter is used for encryption at the migration destination and decryption at the migration source. For details, see **Encryption and Decryption During File Migration**.	AES-256-GCM
	DEK	This parameter is displayed only when **Encryption** is set to **AES-256-GCM**. The key consists of 64 hexadecimal numbers and must be the same as the **DEK** configured during encryption. If the encryption and decryption keys are inconsistent, the system does not report an exception, but the decrypted data is incorrect.	DD0AE00DFECD78BF051BCFDA25BD4E320DB0A7AC75A1F3FC3D3C56A457DCDC1B

Category	Parameter	Description	Example Value
	IV	This parameter is displayed only when **Encryption** is set to **AES-256-GCM**. The initialization vector consists of 32 hexadecimal numbers and must be the same as the **IV** configured during encryption. If the encryption and decryption keys are inconsistent, the system does not report an exception, but the decrypted data is incorrect.	5C91687BA886EDCD12ACBC3FF19A3C3F
	MD5 File Extension	This parameter is displayed only when **File Format** is set to **Binary**. This parameter is used to check whether the files extracted by CDM are consistent with source files. For details, see **MD5 Verification**.	.md5

5.3.3 From HBase/CloudTable

If the source link of a job is an **HBase** or **CloudTable** link, that is, if data is exported from MRS HBase, FusionInsight HBase, CloudTable, or Apache HBase, configure the source job parameters based on **Table 5-9**.

NOTE

1. When you migrate data from CloudTable or HBase, CDM reads the first row of the table as an example of the field list. If the first row of data does not contain all fields of the table, you need to manually add fields.
2. Because HBase is schema-less, CDM cannot obtain the data types. If the data is stored in binary format, CDM cannot parse the data.
3. When data is exported from HBase or CloudTable, because HBase/CloudTable is schema-less storage systems, CDM requires that the source numeric fields be stored in regular decimal format rather than in binary format. For example, the value 100 needs to be stored as **100** rather than **01100100**.

Table 5-9 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Table Name	Name of the HBase table that data will be exported from This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	TBL_2
	Column Families	(Optional) Column families to which the exported data belongs	CF1&CF2
Advanced attributes	Split Rowkey	(Optional) Whether to split a rowkey. The default value is **No**.	Yes
	Rowkey Delimiter	(Optional) Delimiter used to split a rowkey. If this parameter is left empty, the rowkey will not be split.	|

Category	Parameter	Description	Example Value
	Start Time	(Optional) Start time (including the value) for extracting data. The format is *yyyy-MM-dd HH:mm:ss*. Only the data generated at the specified time and later is extracted. This parameter can be set to a macro variable of date and time. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	2019-01-01 20:00:00
	End Time	(Optional) End time (excluding the value) for extracting data. The format is *yyyy-MM-dd HH:mm:ss*. Only the data generated before the time point is extracted. This parameter can be set to a macro variable of date and time. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	2019-02-01 20:00:00

5.3.4 From Hive

If the source link of a job is a **Hive link**, configure the source job parameters based on **Table 5-10**.

Table 5-10 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Database Name	Database name. Click the icon next to the text box. The dialog box for selecting the database is displayed.	default
	Table Name	Hive table name. Click the icon next to the text box. The dialog box for selecting the table is displayed. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	TBL_E

Category	Parameter	Description	Example Value
	Read Mode	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 HDFS mode allows you to use the WHERE condition to filter data or add new fields on the field mapping page.	HDFS
	Use SQL Statement	Whether you can use SQL statements to export data from a relational database	No

Category	Parameter	Description	Example Value

	SQL Statement	When **Use SQL Statement** is set to **Yes**, enter an SQL statement here. CDM exports data based on the SQL statement.
		NOTE
		● SQL statements can only be used to query data. Join and nesting are supported, but multiple query statements are not allowed, for example, **select * from table a; select * from table b**.
		● With statements are not supported.
		● Comments, such as **--** and **/***, are not supported.
		● Addition, deletion, and modification operations are not supported, including but not limited to the following:
		  ● load data
		  ● delete from
		  ● alter table
		  ● create table
		  ● drop table
		  ● into outfile
			select id,name from sqoop.user;

Category	Parameter	Description	Example Value
Advanced attributes	Partition Values	This parameter is displayed when you select the HDFS read mode and click **Show Advanced Attributes**. This parameter indicates extracting the partition of a specified value. The attribute name is the partition name. You can configure multiple values (separated by spaces) or a field value range. The time macro function is supported. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	• Attribute value in the single-value or multi-value filtering scenario: ${dateformat(yyyyMMdd, -1, DAY)} ${dateformat(yyyyMMdd)} • Attribute value in the range filtering scenario: ${value} >= ${dateformat(yyyyMMdd, -7, DAY)} && ${value} < ${dateformat(yyyyMMdd)}

Category	Parameter	Description	Example Value
	WHERE Clause	This parameter is displayed when you select the JDBC read mode and click **Show Advanced Attributes**. This parameter indicates the WHERE clause to be extracted. If this parameter is not set, the entire table is extracted. If the table to be migrated does not contain the fields specified by the WHERE clause, the migration will fail. You can set a date macro variable to extract data generated on a specific date. For details, see **Incremental Migration of Relational Databases**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	age > 18 and age <= 60

NOTE

If the data source is Hive, CDM will automatically partition data using the Hive data partitioning file.

5.3.5 From DLI

If the source link of a job is a **DLI link**, configure the source job parameters based on **Table 5-11**.

Table 5-11 Parameter description

Parameter	Description	Example Value
Resource Queue	Resource queue to which the destination table belongs The default queue of DLI cannot be used for migration jobs. You need to create a SQL queue in DLI.	cdm
Database Name	Name of the database to which data will be written	dli
Table Name	Name of the table to which data will be written	car_detail
Partition	Partition information	year=2020,location=sun

5.3.6 From FTP/SFTP

If the source link of a job is an **FTP or SFTP link**, configure the source job parameters based on **Table 5-12**.

Advanced attributes are optional and not displayed by default. You can click **Show Advanced Attributes** to display them.

Table 5-12 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Source Directory/ File	Directory or file path from which data will be extracted. You can enter a maximum of 50 file paths. By default, the file paths are separated by vertical bars (|). You can also customize a file separator. For details, see **Migration of a List of Files**. Directory from which data is to be migrated. All files (including all nested subdirectories and their subfiles) in the directory will be migrated. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	/ftp/a.csv|/ftp/b.txt
	File Format	Format in which CDM parses data. The options are as follows: • **CSV**: Source files will be migrated to tables after being converted to CSV format. • **Binary**: Files (even not in binary format) will be transferred directly. This format is used to copy data from a file to another. • **JSON**: Source files will be migrated to tables after being converted to JSON format. **NOTE** If the destination is OBS, only the binary format is supported.	CSV
	JSON Type	This parameter is displayed only when **File Format** is set to **JSON**. Type of a JSON object stored in a JSON file. The options are **JSON object** and **JSON array**.	JSON object

Category	Parameter	Description	Example Value
	JSON Reference Node	This parameter is used only when **File Format** is set to **JSON** and **JSON Type** is set to **JSON Object**. CDM parses the data under the JSON node. If the node's corresponding data is a JSON array, the system will extract data from the array in the same pattern. Use periods (.) to separate multi-layer nested JSON nodes.	data.list
Advanced attributes	Use rfc4180 Parser	This parameter is displayed only when **File Format** is set to **CSV**. It specifies whether to use the rfc4180 parser to parse CSV files.	No
	Line Separator	Lind feed character in a file. By default, the system automatically identifies **\n**, **\r**, and **\r\n**. This parameter is displayed only when **File Format** is set to **CSV**.	\n
	Field Delimiter	Character used to separate fields in the file. To set the **Tab** key as the delimiter, set this parameter to **\t**. This parameter is displayed only when **File Format** is set to **CSV**.	,
	Use Quote Character	If you set this parameter to **Yes**, the field delimiters in the encircling symbol are regarded as a part of the string value. Currently, the default encircling symbol of CDM is **"**.	No
	Using Escape Char	If you select **Yes**, the backslash (\) in the data row is used as an escape character. If you select **No**, the backslash (\) in the CSV file will not be escaped. CSV supports only the backslash (\) as the escape character.	Yes
	Use RE to Separate Fields	Whether to use regular expressions to separate fields. If you set this parameter to **Yes**, **Field Delimiter** becomes invalid. This parameter is displayed only when **File Format** is set to **CSV**.	Yes
	Regular Expression	This parameter is available only when **Using RE to separate fields** is set to **Yes**. Regular expression used to separate fields. For details about regular expressions, see **Regular Expressions for Separating Semi-structured Text**.	^(\d.*\d) (\w*) \[(.*) \] ([\w\.]*) (\w.*).*

Category	Parameter	Description	Example Value
	Use First Row as Header	This parameter is displayed only when **File Format** is set to **CSV**. When you migrate a CSV file to a table, CDM writes all data to the table by default. If you set this parameter to **Yes**, CDM uses the first N rows of the CSV file as the heading row and does not write the row to the destination table.	Yes
	Encoding Type	Encoding type, for example, **UTF-8** or **GBK**. You can set the encoding type for text files only. This parameter is invalid when **File Format** is set to **Binary**.	UTF-8
	Compression Format	The options are as follows: • **NONE**: Files in all formats can be transferred. • **GZIP**: Only files in gzip format can be transferred. • **ZIP**: Only files in Zip format can be transferred. • **TAR.GZ**: Files in TAR.GZ format are transferred.	NONE
	Compressed File Suffix	This parameter is displayed when **Compression Format** is not **NONE**. This parameter specifies the extension of the files to be decompressed. The decompression operation is performed only when the file name extension is used in a batch of files. Otherwise, files are transferred in the original format. If you enter ***** or leave the parameter blank, all files are decompressed.	*
	Start Job by Marker File	Whether to start a job by a marker file. A job is only started if there is a marker file for starting the job in the source path. If there is no marker file, the job will be suspended for a period of time specified by **Suspension Period**.	Yes
	File Separator	File separator. If you enter multiple file paths in **Source Directory/Files**, CDM uses the file separator to identify files. The default value is **|**.	|

Category	Parameter	Description	Example Value
	Marker File	Name of the marker file for starting a job. If you specify a marker file, the migration job is executed only when the marker file exists in the source path. The marker file will not be migrated.	ok.txt
	Suspension Period	Waiting period for a marker file. If you set **Start Job by Marker File** to **Yes** but there is no marker file in the source path, the job fails when the suspension period times out. If you set this parameter to **0** and there is no marker file in the source path, the job will fail immediately. Unit: second	10
	Filter Type	Only paths or files that meet the filtering conditions are transferred. The options are **None**, **Wildcard**, and **Regex**. For details, see **Incremental File Migration**.	None
	Directory Filter	If you set **Filter Type** to **Wildcard** or **Regex**, enter a wildcard character to filter paths. The paths that meet the filtering condition are migrated. You can configure multiple paths separated by commas (,). **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	*input,*out
	File Filter	If you set **Filter Type** to **Wildcard** or **Regex**, enter a wildcard character to filter paths. The files that meet the filtering condition are migrated. You can configure multiple files separated by commas (,). **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	*.csv
	Time Filter	If you select **Yes**, files are transferred based on their modification time.	Yes

Category	Parameter	Description	Example Value
	Minimum Timestamp	If you set **Time Filter** to **Yes**, you can specify a point in time for **Minimum Timestamp**, and then only the files modified at or after the specified time are transferred. The time format must be *yyyy-MM-dd HH:mm:ss*. This parameter can be set to a macro variable of date and time. For example, **${timestamp(dateformat(yyyy-MM-dd HH:mm:ss,-90,DAY))}** indicates that only files generated within the latest 90 days are migrated. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	2019-07-01 00:00:00
	Maximum Timestamp	If you set **Time Filter** to **Yes**, you can specify a point in time for **Maximum Timestamp**, and then only the files modified before the specified time are transferred. The time format must be *yyyy-MM-dd HH:mm:ss*. This parameter can be set to a macro variable of date and time. For example, **${timestamp(dateformat(yyyy-MM-dd HH:mm:ss))}** indicates that only the files whose modification time is earlier than the current time are migrated. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	2019-07-30 00:00:00
	Disregard Non-existent Path or File	If this parameter is set to **Yes**, the job can be successfully executed even if the source path does not exist.	No

Category	Parameter	Description	Example Value
	Marker File Type	This parameter is available only when **Start Job by Marker File** is set to **Yes**. • **MARK_DONE**: The migration job is executed only when the marker file exists in the source path. • **MARK_DOING**: The migration job is executed only when the marker file does not exist in the source path.	MARK_DOING
	Whether to skip empty lines	This parameter is available only when **File Format** is set to **CSV**. If a line is empty, it is skipped.	No
	null value	This parameter is available only when **File Format** is set to **Binary**. No string can be used to define a null value in text files. This parameter specifies the string to be identified as a null value.	No
	MD5 File Extension	This parameter is displayed only when **File Format** is set to **Binary**. This parameter is used to check whether the files extracted by CDM are consistent with source files. For details, see **MD5 Verification**.	.md5

5.3.7 From HTTP

If the source link of a job is an HTTP link, configure the source job parameters based on **Table 5-13**. Currently, data can only be exported from the HTTP URLs.

Table 5-13 Parameter description

Parameter	Description	Example Value
File URL	Use the GET method to obtain data from the HTTP/HTTPS URL. These connectors are used to read files with an HTTP/HTTPS URL, such as reading public files on the third-party object storage system and web disks.	https://bucket.obs.myhuaweicloud.com/object-key

Parameter	Description	Example Value
Pull List File	If this parameter is set to **Yes**, the system pulls the files corresponding to the URLs in the text file to be uploaded and stores them on OBS. The text file records the file paths on HDFS.	Yes
OBS Link of List File	Select an existing OBS link.	obs_link
OBS Bucket of entries files	Name of the OBS bucket that stores the text file	obs-cdm
Path/ Directory of entries files	Custom OBS directories that store the text file. Use slashes (/) to separate different directories.	test1
File Format	CDM supports **Binary** only, which indicates that files (even not in binary format) will be directly transferred.	Binary
Compression Format	Compression format of the source files. The options are as follows: ● **NONE**: Files in all formats can be transferred. ● **GZIP**: Only files in gzip format can be transferred. ● **ZIP**: Only files in Zip format can be transferred. ● **TAR.GZ**: Files in TAR.GZ format are transferred.	NONE
Compressed File Suffix	This parameter is displayed when **Compression Format** is not **NONE**. This parameter specifies the extension of the files to be decompressed. The decompression operation is performed only when the file name extension is used in a batch of files. Otherwise, files are transferred in the original format. If you enter ***** or leave the parameter blank, all files are decompressed.	*
File Separator	File separator. When multiple files are transferred, CDM uses the file separator to identify files. The default value is **|**. This parameter is not displayed if **Pull List File** is set to **Yes**.	|

Parameter	Description	Example Value
Query Parameter	• If you set this parameter to **Yes**, the name of the objects uploaded to OBS does not include the **query** parameter. • If you set this parameter to **No**, the name of the objects uploaded to OBS includes the **query** parameter.	No
Disregard Non-existent Path or File	If this is set to **Yes**, the job can be successfully executed even if the source path does not exist.	No
MD5 File Extension	This parameter is used to check whether the files extracted by CDM are consistent with source files. For details, see **MD5 Verification**.	.md5

5.3.8 From PostgreSQL/SQL Server

If the source link of a job is an RDS for PostgreSQL, RDS for SQL Server, PostgreSQL, or Microsoft SQL Server link, configure the source job parameters based on **Table 5-14**.

Table 5-14 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Use SQL Statement	Whether you can use SQL statements to export data from a relational database	No

Category	Parameter	Description	Example Value

	SQL Statement	When **Use SQL Statement** is set to **Yes**, enter an SQL statement here. CDM exports data based on the SQL statement.
		NOTE
		• SQL statements can only be used to query data. Join and nesting are supported, but multiple query statements are not allowed, for example, **select * from table a; select * from table b**.
		• With statements are not supported.
		• Comments, such as **--** and **/***, are not supported.
		• Addition, deletion, and modification operations are not supported, including but not limited to the following:
		  • load data
		  • delete from
		  • alter table
		  • create table
		  • drop table
		  • into outfile
			select id,name from sqoop.user;

	Schema/ Tablespace	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. 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. The examples are as follows: • **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.	SCHEMA_E

Category	Parameter	Description	Example Value
	Table Name	Name of the table 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 the table or directly enter a table name. If the desired table is not displayed, check whether the table exists or whether the login account has the permission to query metadata. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*). **NOTE** The table name can contain wildcard characters (*), which is used to export all tables whose names start with a certain prefix or end with a certain suffix. The number and types of fields in the tables must be the same. The examples are as follows: • **table*** indicates that all tables whose names starting with **table** are exported. • ***table** indicates that all tables whose names ending with **table** are exported. • ***table*** indicates that all tables whose names containing **table** are exported.	table

Category	Parameter	Description	Example Value
Advanced attributes	Partition Column	This parameter is displayed when **Use SQL Statement** is set to **No**, indicating that a field used to split data during data extraction. CDM splits a job into multiple tasks based on this field and executes the tasks concurrently. Fields with data distributed evenly are used, such as the sequential number field. Click the icon next to the text box to go to the page for selecting a field or directly enter a field. **NOTE** The following types of partition columns are supported: TINYINT, SMALLINT, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, NUMERIC, DECIMAL, BIT, BOOLEAN, DATE, TIME, and TIMESTAMP. It is recommended that the partition column have an index.	id
	Where Clause	WHERE clause used to specify the data extraction range. This parameter is displayed when **Use SQL Statement** is set to **No**. If this parameter is not set, the entire table is extracted. You can set a date macro variable to extract data generated on a specific date. For details, see **Incremental Migration of Relational Databases**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	DS='${dateformat(yyyy-MM-dd,-1,DAY)}'
	Null in Partition Column	Whether the partition column can contain null values	Yes

Category	Parameter	Description	Example Value
	Extract by Partition	Data can be extracted from each partition in the partitioned table. If this function is enabled, you can configure **Table Partition** to specify specific table partitions from which data is extracted. • This function does not support non-partitioned tables. • This parameter can be configured only when the migration source is a PostgreSQL database. • The database user must have the **SELECT** permission on the system views **dba_tab_partitions** and **dba_tab_subpartitions**.	No
	Split Job	If this parameter is set to **Yes**, the job is split into multiple subjobs based on the value of **Job Split Field**, and the subjobs are executed concurrently. **NOTE** This parameter and parameters *Job Split Field*, *Minimum Split Field Value*, *Maximum Split Field Value*, and *Number of subjobs* are available only when the destination link is a DLI or Hive link.	Yes
	Job Split Field	Field used to split a job into multiple subjobs for concurrent execution. This parameter is available when **Split Job** is set to **Yes**.	-
	Minimum Split Field Value	Minimum value of **Job Split Field** during data extraction. This parameter is available when **Split Job** is set to **Yes**.	-
	Maximum Split Field Value	Maximum value of **Job Split Field** during data extraction. This parameter is available when **Split Job** is set to **Yes**.	-
	Number of subjobs	Number of subjobs split from a job for concurrent execution based on the data range specified by the minimum and maximum values of **Job Split Field**. This parameter is available when **Split Job** is set to **Yes**.	-

5.3.9 From DWS

If the source link of a job is a **DWS link**, configure the source job parameters based on **Table 5-15**.

Table 5-15 Parameter description

Type	Parameter	Description	Example Value
Basic parameters	Use SQL Statement	Whether you can use SQL statements to export data from a relational database	No

	SQL Statement	When **Use SQL Statement** is set to **Yes**, enter an SQL statement here. CDM exports data based on the SQL statement.
		NOTE
		• SQL statements can only be used to query data. Join and nesting are supported, but multiple query statements are not allowed, for example, **select * from table a; select * from table b**.
		• With statements are not supported.
		• Comments, such as **--** and **/***, are not supported.
		• Addition, deletion, and modification operations are not supported, including but not limited to the following:
		  • load data
		  • delete from
		  • alter table
		  • create table
		  • drop table
		  • into outfile
			select id,name from sqoop.user;

Type	Parameter	Description	Example Value
	Schema/Tablespace	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. 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. Examples: • **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.	SCHEMA_E

Type	Parameter	Description	Example Value
	Table Name	Name of the table 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 the table or directly enter a table name. If the desired table is not displayed, check whether the table exists or whether the login account has the permission to query metadata. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*). **NOTE** The table name can contain wildcard characters (*), which is used to export all tables whose names start with a certain prefix or end with a certain suffix. The number and types of fields in the tables must be the same. The examples are as follows: • **table*** indicates that all tables whose names starting with **table** are exported. • ***table** indicates that all tables whose names ending with **table** are exported. • ***table*** indicates that all tables whose names containing **table** are exported.	table

Type	Parameter	Description	Example Value
Advanced attributes	WHERE Clause	WHERE clause used to specify the data extraction range. This parameter is displayed when **Use SQL Statement** is set to **No**. If this parameter is not set, the entire table is extracted. You can set a date macro variable to extract data generated on a specific date. For details, see **Incremental Migration of Relational Databases**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	DS='${dateformat(yyyy-MM-dd,-1,DAY)}'
	Partition Column	This parameter is displayed when **Use SQL Statement** is set to **No**, indicating a field used to split data during data extraction. CDM splits a job into multiple tasks based on this field and executes the tasks concurrently. Fields with data distributed evenly are used, such as the sequential number field. Click the icon next to the text box to go to the page for selecting a field or directly enter a field. **NOTE** The following types of partition columns are supported: TINYINT, SMALLINT, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, NUMERIC, DECIMAL, BIT, BOOLEAN, DATE, TIME, and TIMESTAMP. It is recommended that the partition column have an index.	id
	Null in Partition Column	Whether the partition column can contain null values	Yes
	Split Job	If this parameter is set to **Yes**, the job is split into multiple subjobs based on the value of **Job Split Field**, and the subjobs are executed concurrently. **NOTE** This parameter and parameters *Job Split Field*, *Minimum Split Field Value*, *Maximum Split Field Value*, and *Number of subjobs* are available only when the destination link is a DLI or Hive link.	Yes

Type	Parameter	Description	Example Value
	Job Split Field	Field used to split a job into multiple subjobs for concurrent execution. This parameter is available when **Split Job** is set to **Yes**.	-
	Minimum Split Field Value	Minimum value of **Job Split Field** during data extraction. This parameter is available when **Split Job** is set to **Yes**.	-
	Maximum Split Field Value	Maximum value of **Job Split Field** during data extraction. This parameter is available when **Split Job** is set to **Yes**.	-
	Number of subjobs	Number of subjobs split from a job for concurrent execution based on the data range specified by the minimum and maximum values of **Job Split Field**. This parameter is available when **Split Job** is set to **Yes**.	-

5.3.10 From SAP HANA

Table 5-16 lists the job parameters when the source link is a SAP HANA link.

Table 5-16 Parameter description

Type	Parameter	Description	Example Value
Basic parameters	Use SQL Statement	Whether you can use SQL statements to export data from a relational database	No

Type	Parameter	Description	Example Value

	SQL Statement	When **Use SQL Statement** is set to **Yes**, enter an SQL statement here. CDM exports data based on the SQL statement.
		NOTE
		• SQL statements can only be used to query data. Join and nesting are supported, but multiple query statements are not allowed, for example, **select * from table a; select * from table b**.
		• With statements are not supported.
		• Comments, such as **--** and **/***, are not supported.
		• Addition, deletion, and modification operations are not supported, including but not limited to the following:
		  • load data
		  • delete from
		  • alter table
		  • create table
		  • drop table
		  • into outfile
			select id,name from sqoop.user;

	Schema/ Tablespace	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. 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. Examples: • **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.	SCHEMA_E

Type	Parameter	Description	Example Value
	Table Name	Name of the table 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 the table or directly enter a table name. If the desired table is not displayed, check whether the table exists or whether the login account has the permission to query metadata. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*). **NOTE** The table name can contain wildcard characters (*), which is used to export all tables whose names start with a certain prefix or end with a certain suffix. The number and types of fields in the tables must be the same. The examples are as follows: • **table*** indicates that all tables whose names starting with **table** are exported. • ***table** indicates that all tables whose names ending with **table** are exported. • ***table*** indicates that all tables whose names containing **table** are exported.	table

Type	Parameter	Description	Example Value
Advanced attributes	WHERE Clause	WHERE clause used to specify the data extraction range. This parameter is displayed when **Use SQL Statement** is set to **No**. If this parameter is not set, the entire table is extracted. You can set a date macro variable to extract data generated on a specific date. For details, see **Incremental Migration of Relational Databases**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	DS='${dateformat(yyyy-MM-dd,-1,DAY)}'
	Partition Column	This parameter is displayed when **Use SQL Statement** is set to **No**, indicating a field used to split data during data extraction. CDM splits a job into multiple tasks based on this field and executes the tasks concurrently. Fields with data distributed evenly are used, such as the sequential number field. Click the icon next to the text box to go to the page for selecting a field or directly enter a field. **NOTE** The following types of partition columns are supported: TINYINT, SMALLINT, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, NUMERIC, DECIMAL, BIT, BOOLEAN, DATE, TIME, and TIMESTAMP. It is recommended that the partition column have an index.	id

5.3.11 From MySQL

If the source link of a job is an **RDS for MySQL or MySQL link**, configure the source job parameters based on **Table 5-17**.

Table 5-17 Parameter description

Parameter	Description	Example Value
Use SQL Statement	Whether you can use SQL statements to export data from a relational database	No

Parameter	Description	Example Value

SQL Statement	When **Use SQL Statement** is set to **Yes**, enter an SQL statement here. CDM exports data based on the SQL statement.
	NOTE
	• SQL statements can only be used to query data. Join and nesting are supported, but multiple query statements are not allowed, for example, **select * from table a; select * from table b**.
	• With statements are not supported.
	• Comments, such as **--** and **/***, are not supported.
	• Addition, deletion, and modification operations are not supported, including but not limited to the following:
	  • load data
	  • delete from
	  • alter table
	  • create table
	  • drop table
	  • into outfile
		select id,name from sqoop.user;

Schema/ Tablespace	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. If the desired schema or tablespace is not displayed, check whether the login account has the permissions required to query metadata. This parameter can be set to a regular expression to export all databases that meet the rule.	SCHEMA_E

Parameter	Description	Example Value
Table Name	Name of the table 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 the table or directly enter a table name. If the desired table is not displayed, check whether the table exists or whether the login account has the permission to query metadata. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. This parameter can be set to a regular expression to export all databases that meet the rule. For example, if **Table Name** is set to *user_[0-9]{1,2}*, tables from **user_0** to **user_9** and from **user_00** to **user_99** are matched. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	table
Partition Column	This parameter is displayed when **Use SQL Statement** is set to **No**, indicating a field used to split data during data extraction. CDM splits a job into multiple tasks based on this field and executes the tasks concurrently. Fields with data distributed evenly are used, such as the sequential number field. Click the icon next to the text box to go to the page for selecting a field or directly enter a field. **NOTE** The following types of partition columns are supported: TINYINT, SMALLINT, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, NUMERIC, DECIMAL, BIT, BOOLEAN, DATE, TIME, and TIMESTAMP. It is recommended that the partition column have an index.	id

Parameter	Description	Example Value
Where Clause	WHERE clause used to specify the data extraction range. This parameter is displayed when **Use SQL Statement** is set to **No**. If this parameter is not set, the entire table is extracted. You can set a date macro variable to extract data generated on a specific date. For details, see **Incremental Migration of Relational Databases**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	DS='${dateformat(yyyy-MM-dd,-1,DAY)}'
Null in Partition Column	Whether the partition column can contain null values	Yes
Split Job	If this parameter is set to **Yes**, the job is split into multiple subjobs based on the value of **Job Split Field**, and the subjobs are executed concurrently. **NOTE** This parameter and parameters *Job Split Field*, *Minimum Split Field Value*, *Maximum Split Field Value*, and *Number of subjobs* are available only when the destination link is a DLI or Hive link.	Yes
Job Split Field	Field used to split a job into multiple subjobs for concurrent execution. This parameter is available when **Split Job** is set to **Yes**.	-
Minimum Split Field Value	Minimum value of **Job Split Field** during data extraction. This parameter is available when **Split Job** is set to **Yes**.	-
Maximum Split Field Value	Maximum value of **Job Split Field** during data extraction. This parameter is available when **Split Job** is set to **Yes**.	-
Number of subjobs	Number of subjobs split from a job for concurrent execution based on the data range specified by the minimum and maximum values of **Job Split Field**. This parameter is available when **Split Job** is set to **Yes**.	-

Parameter	Description	Example Value
Extract by Partition	When data is exported from a MySQL database, data can be extracted from each partition in the partitioned table. If this function is enabled, you can configure **Table Partition** to specify specific MySQL table partitions from which data is extracted. • This function does not support non-partitioned tables. • The database user must have the **SELECT** permission on the system views **dba_tab_partitions** and **dba_tab_subpartitions**.	No

5.3.12 From Oracle

If the source link of a job is an **Oracle link**, configure the source job parameters based on **Table 5-18**.

Table 5-18 Parameter description

Parameter	Description	Example Value
Use SQL Statement	Whether you can use SQL statements to export data from a relational database	No

SQL Statement	When **Use SQL Statement** is set to **Yes**, enter an SQL statement here. CDM exports data based on the SQL statement.
	NOTE
	• SQL statements can only be used to query data. Join and nesting are supported, but multiple query statements are not allowed, for example, **select * from table a; select * from table b**.
	• With statements are not supported.
	• Comments, such as **--** and **/***, are not supported.
	• Addition, deletion, and modification operations are not supported, including but not limited to the following:
	  • load data
	  • delete from
	  • alter table
	  • create table
	  • drop table
	  • into outfile
		select id,name from sqoop.user;

Parameter	Description	Example Value
Schema/ Tablespace	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. 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.	SCHEMA_E

Parameter	Description	Example Value
Table Name	Name of the table 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 the table or directly enter a table name. If the desired table is not displayed, check whether the table exists or whether the login account has the permission to query metadata. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*). **NOTE** The table name can contain wildcard characters (*), which is used to export all tables whose names start with a certain prefix or end with a certain suffix. The number and types of fields in the tables must be the same. The examples are as follows: • **table*** indicates that all tables whose names starting with **table** are exported. • ***table** indicates that all tables whose names ending with **table** are exported. • ***table*** indicates that all tables whose names containing **table** are exported.	table

Parameter	Description	Example Value
Partition Column	This parameter is displayed when **Extract by Partition** is set to **No**, indicating a field used to split data during data extraction. CDM splits a job into multiple tasks based on this field and executes the tasks concurrently. Fields with data distributed evenly are used, such as the sequential number field. Click the icon next to the text box to go to the page for selecting a field or directly enter a field. **NOTE** The following types of partition columns are supported: TINYINT, SMALLINT, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, NUMERIC, DECIMAL, BIT, BOOLEAN, DATE, TIME, and TIMESTAMP. It is recommended that the partition column have an index.	id
Where Clause	WHERE clause used to specify the data extraction range. This parameter is displayed when **Use SQL Statement** is set to **No**. If this parameter is not set, the entire table is extracted. You can set a date macro variable to extract data generated on a specific date. For details, see **Incremental Migration of Relational Databases**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	DS='${dateformat(yyyy-MM-dd,-1,DAY)}'
Null in Partition Column	Whether the partition field can contain null values. This parameter is displayed when **Extract by Partition** is set to **No**.	Yes
Extract by Partition	When data is exported from an Oracle database, data can be extracted from each partition in the partitioned table. If this function is enabled, you can configure **Table Partition** to specify specific Oracle table partitions from which data is extracted. • This function does not support non-partitioned tables. • The database user must have the **SELECT** permission on the system views **dba_tab_partitions** and **dba_tab_subpartitions**.	No

Parameter	Description	Example Value
Table Partition	Oracle table partition from which data is migrated. Separate multiple partitions with ampersands (&). If you do not set this parameter, all partitions will be migrated. If there is a subpartition, enter the partition in the *Partition.Subpartition* format, for example, **P2.SUBP1**.	P0&P1&P2.SUBP1&P2.SUBP3
Split Job	If this parameter is set to **Yes**, the job is split into multiple subjobs based on the value of **Job Split Field**, and the subjobs are executed concurrently. **NOTE** This parameter and parameters *Job Split Field*, *Minimum Split Field Value*, *Maximum Split Field Value*, and *Number of subjobs* are available only when the destination link is a DLI or Hive link.	Yes
Job Split Field	Field used to split a job into multiple subjobs for concurrent execution. This parameter is available when **Split Job** is set to **Yes**.	-
Minimum Split Field Value	Minimum value of **Job Split Field** during data extraction. This parameter is available when **Split Job** is set to **Yes**.	-
Maximum Split Field Value	Maximum value of **Job Split Field** during data extraction. This parameter is available when **Split Job** is set to **Yes**.	-
Number of subjobs	Number of subjobs split from a job for concurrent execution based on the data range specified by the minimum and maximum values of **Job Split Field**. This parameter is available when **Split Job** is set to **Yes**.	-

NOTE

When an Oracle database is the migration source, if **Partitioning Field** or **Extract by Partition** is not configured, CDM automatically uses the ROWIDs to partition data.

5.3.13 From a Database Shard

If the source link of a job is a **database shard link**, configure the source job parameters based on **Table 5-19**.

Table 5-19 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Schema/ Tablespace	Indicates the name of the schema or tablespace from which data is to be extracted. Click the icon next to the text box to go to the page for selecting a schema or tablespace. During a sharded link job, the tablespace corresponding to the first backend link is displayed by default. You can also enter a schema or tablespace name. If the desired schema or tablespace is not displayed, check whether the login account has the permissions required to query metadata. This parameter can be set to a regular expression to export all databases that meet the rule. For example, if **Table Name** is set to *user_[0-9]{1,2}*, tables from **user_0** to **user_9** and from **user_00** to **user_99** are matched.	SCHEMA_E

Category	Parameter	Description	Example Value
	Table Name	Indicates the name of the table from which data is to be extracted. Click the icon next to the text box to go to the page for selecting the table or directly enter a table name. If the desired table is not displayed, check whether the table exists or whether the login account has the permission to query metadata. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. This parameter can be set to a regular expression to export all databases that meet the rule. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	table
Advanced attributes	WHERE Clause	Specifies the data extraction range. If this parameter is not set, the entire table is extracted. You can set a date macro variable to extract data generated on a specific date. For details, see **Incremental Migration of Relational Databases**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	DS='${dateformat(yyyy-MM-dd,-1,DAY)}'

NOTE

- If the **Source Link Name** is the backend link of the sharded link, the job is a common MySQL job.
- When creating a job whose source end is a sharded link, you can add a custom field with the sample value of **${custom(host)}** to the source field during field mapping. This field is used to view the data source of the table after the data of multiple tables across databases is migrated to the same table. The following sample values are supported:
  - ${custom(host)}
  - ${custom(database)}
  - ${custom(fromLinkName)}
  - ${custom(schemaName)}
  - ${custom(tableName)}

5.3.14 From MongoDB/DDS

When you migrate MongoDB or DDS data, CDM reads the first row of the collection as an example of the field list. If the first row of data does not contain all fields of the collection, you need to manually add fields.

If the source link of a job is a **MongoDB link**, that is, if data is exported from an on-premises MongoDB or DDS, configure the source job parameters based on **Table 5-20**.

Table 5-20 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Database Name	Name of the database from which data will be migrated	mongodb
	Collection Name	Collection name, similar to the table name of a relational database. Click the icon next to the text box to go to the page for selecting the collection or directly enter a collection name. If the desired table is not displayed, check whether the table exists or whether the login account has the permission to query metadata.	COLLECTION

Category	Parameter	Description	Example Value
Advanced attributes	Filter Condition	Conditions for filtering documents. CDM migrates only the data that meets the filter conditions. The examples are as follows: 1. Filter by expression: **{'last_name': 'Smith'}** indicates that all files whose **last_name** value is **Smith** are queried. 2. Filter by parameter: **{ x : "john" }, { z : 1 }** indicates that all **z** fields whose **x** is **john** are queried. 3. Filter by condition: **{ "field" : { $gt: 5 } }** indicates that the **field** values greater than 5 are queried. 4. Filter by time macro: {"ts":{$gte:ISODate("${dateformat(yyyy-MM-dd'T'HH:mm:ss.SSS'Z',-1,HOUR)}")}} indicates that the values greater than those after time macro conversion in the **ts** field are queried.	{'last_name': 'Smith'}

5.3.15 From Redis

The Redis service of the third-party cloud cannot serve as the migration source. However, the Redis set up in the on-premises data center or on the ECS can be the migration source and destination.

If the source link of a job is an on-premises Redis link, configure the source job parameters based on **Table 5-21**.

Table 5-21 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Redis Key Prefix	Key prefix, which is similar to the table name of a relational database	TABLE
	Value Storage Type	The options are as follows: • **String**: without column name, such as **value1,value2** • **Hash**: with column name, such as **column1=value1,column2=value2**	String

Category	Parameter	Description	Example Value
Advanced attributes	Key Delimiter	Character used to separate table names and column names of a relational database	_
	Value Delimiter	Character used to separate columns when the storage type is string	;
	Same Field	This parameter is displayed when **Value Storage Type** is set to **Hash**. The hash key contains the same field.	Yes

5.3.16 From DIS

The data in the message body is a record in CSV format that supports multiple delimiters. Messages cannot be parsed in binary or other formats.

If the source link of a job is a **DIS link**, configure the source job parameters based on **Table 5-22**.

Table 5-22 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	DIS Stream	DIS stream name	dis
	Permanent Running	Whether a job runs permanently. If a job is set to run for a long time, the job will fail if the DIS system is interrupted.	Yes
	DIS Partition ID	ID of the DIS partition. You can enter multiple partition IDs separated by commas (,).	0,1,2
	Offset	Initial offset when data is pulled from DIS • **Latest**: Maximum offset, indicating that the latest data will be extracted. • **From last stop**: Data read will start from which the last read ended. • **Earliest**: Minimum offset, indicating that the earliest data will be extracted.	Latest
	Application Name	Unique identifier of the consumer application to be used. If no application exists, CDM creates one automatically.	cdm

Category	Parameter	Description	Example Value
	Data Format	Format used for parsing data. The options are as follows: • **Binary**: Data is transferred directly. It is not converted to another format. This setting is suitable for file migration. • **CSV**: Source data will be migrated after being converted in CSV format. • **JSON**: Source data will be migrated after being converted in JSON format.	Binary
	Field Delimiter	This parameter is displayed when **Data Format** is set to **CSV**. The default value is comma (,). To set the **Tab** key as the delimiter, set this parameter to **\t**.	,
	Record Delimiter	This parameter is displayed when **Data Format** is set to **CSV** or **JSON**. It is used to separate each two records.	,
Advanced attributes	Max. Poll Records	(Optional) Maximum number of records per poll	100

5.3.17 From Kafka/DMS Kafka

If the source link of a job is a **Kafka link** or **DMS Kafka link**, configure the source job parameters based on **Table 5-23**.

Table 5-23 Parameter description

Type	Parameter	Description	Example Value
Basic parameters	Topics	One or more topics can be entered.	est1,est2

Type	Parameter	Description	Example Value
	Data Format	Format used for parsing data. The options are as follows: • **Binary**: Data is transferred directly. It is not converted to another format. This setting is suitable for file migration. • **CSV**: Source data will be migrated after being converted in CSV format. • **JSON**: Source data will be migrated after being converted in JSON format. • **CDC (DRS)**: Source data will be migrated after being converted in DRS format. • **CDC (JSON)**: Source data will be migrated after being converted in JSON format. • **CDC (DRS_AVRO)**: Source data will be migrated after being converted in DRS_AVRO format. • **CDC (DRS_JSON)**: Source data will be migrated after being converted in DRS_JSON format.	Binary
	Offset	Initial offset parameter • **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	Latest
	Data Extraction Timeout Duration	Maximum duration (minutes) of data extraction. For example, a job scheduled daily needs a sufficient duration to extract the data generated by the topic every day.	60
	Suspension Period	If the value is set to 60 and no data is returned within 60s after the consumer requests data extraction from Kafka (generally because all the data in the topic has been read or the network or Kafka cluster is unavailable), the task will stop immediately. Otherwise, the system will retry reading data.	60
	Consumer Group ID	Consumer group ID If you export data from DMS Kafka, enter any value for Kafka Platinum but a valid consumer group ID for Kafka Basic.	sumer-group

Type	Parameter	Description	Example Value
	Start Time	This parameter is required when **Offset** is set to **Time Range**. It specifies the start time for pulling data, including the data at the specified time point.	2020-12-20 12:00:00
	End Time	This parameter is required when **Offset** is set to **Time Range**. It specifies the end time for pulling data, excluding the data at the specified time point.	2020-12-20 20:00:00
	Field Delimiter	This parameter is required when **Data Format** is set to **CSV**. The default value is space. To set the **Tab** key as the delimiter, set this parameter to **\t**.	,
	Record Delimiter	This parameter is required when **Data Format** is set to **CSV** or **JSON**. The default value is space. To set the **Tab** key as the delimiter, set this parameter to **\t**.	,
Advanced parameters	UseConfigFile	This parameter is required when **Data Format** is set to **CDC**. It is used to configure OBS files.	No
	OBS Link	Select an OBS link.	obs_link
	OBS Bucket	Select an OBS bucket.	obs_test
	Config File	Select the OBS configuration file.	/obs/config.csv
	Max. Poll Records	(Optional) Maximum number of records per poll	100
	Max. Poll Interval	(Optional) Maximum interval between polls (seconds)	100
	Notice Topic	Topic for sending notification data. If the data format is CDC, the notification content is the names of the generated files.	notice

5.3.18 From Elasticsearch or CSS

If the source link of a job is a link described in **Link to Elasticsearch** or **Link to CSS**, configure the source job parameters based on **Table 5-24**.

Table 5-24 Job parameters when Elasticsearch or CSS is the source

Category	Parameter	Description	Example Value
Basic parameters	Index	Elasticsearch index, which is similar to the name of a relational database. The index name can contain only lowercase letters.	index
	Type	Elasticsearch type, which is similar to the table name of a relational database. The type name can contain only lowercase letters. **NOTE** Elasticsearch 7.x and later versions do not support custom types. Instead, only the **_doc** type can be used. In this case, this parameter does not take effect even if it is set.	_doc
Advanced attributes	Split Nested Field	(Optional) Whether to split the JSON content of the nested fields. For example, **a:{ b:{ c:1, d:{ e:2, f:3 } } }** can be split into **a.b.c**, **a.b.d.e**, and **a.b.d.f**.	No

Category	Parameter	Description	Example Value
	Filter Conditions	(Optional) CDM migrates only the data that meets the filter conditions. • Currently, only the query string (q syntax) of Elasticsearch can be used to filter source data. The q syntax is used in the following way: – In exact match, the ***column*:*data*** format is used to match and filter data. ***column*** indicates the field name, and ***data*** indicates the query condition, for example, **last_name:Smith**. In addition, if ***data*** is a string containing spaces, it must be enclosed in double quotation marks. If ***column*** is not specified, all fields will be matched by ***data***. – Multiple query conditions can be combined with connection words. The format is ***column1*:*data1* AND *column2*:*data2***. The connection words can be **AND**, **OR**, or **NOT**. They must be in uppercase, and there must be a space before and after each connection word. Example: **first_name:Alec AND last_name:John** – In range matching, you can directly use a condition expression to filter data. The expression is in ***column*:>*data*** format. The operator can be **>**, **>=**, **<**, or **<=**. An example is **time:>=1636905600000 AND time:<1637078400000**. It can also be used together with a macro variable of date and time, for example, **createTime:>=${timestamp(dateformat(yyyyMMdd,-1,DAY))} AND createTime:< ${timestamp(dateformat(yyyyMMdd))}**. – In range matching, you can also use the range syntax to filter data. The format is ***column*:{*data1* TO *data2*}**. **{** and **}** indicate that a value is not included. **[** and **]** indicate that a	last_name:Smith

Category	Parameter	Description	Example Value
		value is included. **TO** must be capitalized, and there must be a space before and after it. * indicates all data. For example, **time:{1636992000000 TO *]** filters out all the data greater than 1636992000000 in the **time** field. It can also be used together with a macro variable of date and time, for example, **createTime:[${timestamp(dateformat(yyyyMMdd,-1,DAY))} TO ${timestamp(dateformat(yyyyMMdd))}}**. • Source data cannot be filtered using the query domain-specific language (DSL) of Elasticsearch.	
	Extract Meta-field	Whether to extract index meta-fields. For example, _index, _type, _id, and _score.	Yes
	Page size	Elasticsearch page size	1000
	ScrollId Time Out	During a scroll query using Elasticsearch, a **scroll_id** is recorded. When the query times out or is complete, the recorded **srcoll_id** will be cleared. You can set this parameter to specify the timeout duration.	5

5.3.19 From MRS Hudi

If the source link of a job is an **MRS Hudi link**, configure the source job parameters based on **Table 5-25**.

Table 5-25 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Source Link Name	MRS Hudi link	hudi_from_cdm
	Database Name	Database name. Click the icon next to the text box. The dialog box for selecting the database is displayed.	default

Category	Parameter	Description	Example Value
	Table Name	Hudi table name. Click the icon next to the text box. The dialog box for selecting the table is displayed. You can set a macro variable of date and time, and a path name can contain multiple macro variables. You can use macro variables of date and time in a scheduled job to synchronize incremental data periodically. For details, see **Using Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	TBL_E
Advanced attributes	Where Clause	This parameter indicates the where clause to be extracted. If this parameter is not set, the entire table is extracted. If the table to be migrated does not contain the fields specified by the where clause, the migration will fail. You can set a macro variable of date and time to extract the data generated on a specific date. For details, see **Incremental Migration of Relational Databases**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	age > 18 and age <= 60

5.3.20 From MRS ClickHouse

If the source link of a job is an **MRS ClickHouse link**, configure the source job parameters based on **Table 5-26**.

Table 5-26 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Source Link Name	MRS ClickHouse link	ck_from_cdm
	Schema/ Tablespace	Click the icon next to the text box to go to the page for selecting a schema or directly enter a schema or tablespace. If the desired schema or tablespace is not displayed, check whether the login account has the permissions required to query metadata. **NOTE** This parameter can be set to a regular expression to export all databases that meet the rule.	default
	Table Name	Click the icon next to the text box to go to the page for selecting the table or directly enter a table name. If the desired table is not displayed, check whether the table exists or whether the login account has the permission to query metadata. **NOTE** This parameter can be set to a regular expression to export all databases that meet the rule.	TBL_E
Advanced attributes	WHERE Clause	This parameter indicates the WHERE clause to be extracted. If this parameter is not set, the entire table is extracted. If the table to be migrated does not contain the fields specified by the WHERE clause, the migration will fail. You can set a date macro variable to extract data generated on a specific date. For details, see **Incremental Migration of Relational Databases**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	age > 18 and age <= 60

5.3.21 From a Dameng Database

If the source link of a job is a Dameng database link, configure the source job parameters based on **Table 5-27**.

Table 5-27 Parameter description

Type	Parameter	Description	Example Value
Basic parameters	Use SQL Statement	Whether you can use SQL statements to export data from a relational database	No

	SQL Statement	When **Use SQL Statement** is set to **Yes**, enter an SQL statement here. CDM exports data based on the SQL statement.
		NOTE
		• SQL statements can only be used to query data. Join and nesting are supported, but multiple query statements are not allowed, for example, **select * from table a; select * from table b**.
		• With statements are not supported.
		• Comments, such as **--** and **/***, are not supported.
		• Addition, deletion, and modification operations are not supported, including but not limited to the following:
		  • load data
		  • delete from
		  • alter table
		  • create table
		  • drop table
		  • into outfile
			select id,name from sqoop.user;

Type	Parameter	Description	Example Value
	Schema/Tablespace	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. 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.	SCHEMA_E

Type	Parameter	Description	Example Value
	Table Name	Name of the table 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 the table or directly enter a table name. If the desired table is not displayed, check whether the table exists or whether the login account has the permission to query metadata. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*). **NOTE** The table name can contain wildcard characters (*), which is used to export all tables whose names start with a certain prefix or end with a certain suffix. The number and types of fields in the tables must be the same. The examples are as follows: • **table*** indicates that all tables whose names starting with **table** are exported. • ***table** indicates that all tables whose names ending with **table** are exported. • ***table*** indicates that all tables whose names containing **table** are exported.	table

Type	Parameter	Description	Example Value
Advanced attributes	Partition Column	This parameter is displayed when **Use SQL Statement** is set to **No**, indicating that a field used to split data during data extraction. CDM splits a job into multiple tasks based on this field and executes the tasks concurrently. Fields with data distributed evenly are used, such as the sequential number field. Click the icon next to the text box to go to the page for selecting a field or directly enter a field. **NOTE** • The following types of partition columns are supported: CHAR, VARCHAR, LONGVARCHAR, TINYINT, SMALLINT, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, NUMERIC, DECIMAL, BIT, BOOLEAN, DATE, TIME, and TIMESTAMP. It is recommended that the partition column have an index. • If the partition column type is CHAR, VARCHAR, or LONGVARCHAR, the column value cannot contain characters other than those in the ASCII character code table or Chinese characters.	id
	Where Clause	Where clause used to specify the data extraction range. This parameter is displayed when **Use SQL Statement** is set to **No**. If this parameter is not set, the entire table is extracted. You can set a date macro variable to extract data generated on a specific date. For details, see **Incremental Migration of Relational Databases**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	DS='${dateformat(yyyy-MM-dd,-1,DAY)}'
	Null in Partition Column	Whether the partition column can contain null values	Yes

5.3.22 From a ShenTong Database

If the source link of a job is a ShenTong database link, configure the source job parameters based on **Table 5-28**.

Table 5-28 Parameter description

Type	Parameter	Description	Example Value
Basic parameters	Use SQL Statement	Whether you can use SQL statements to export data from a relational database	No

	SQL Statement	When **Use SQL Statement** is set to **Yes**, enter an SQL statement here. CDM exports data based on the SQL statement.
		NOTE
		• SQL statements can only be used to query data. Join and nesting are supported, but multiple query statements are not allowed, for example, **select * from table a; select * from table b**.
		• With statements are not supported.
		• Comments, such as **--** and **/***, are not supported.
		• Addition, deletion, and modification operations are not supported, including but not limited to the following:
		  • load data
		  • delete from
		  • alter table
		  • create table
		  • drop table
		  • into outfile
		• Quote characters take effect only for SQL statements generated in the database table configuration, and cannot be added to custom SQL statements.
			select id,name from sqoop.user;

Type	Parameter	Description	Example Value
	Schema/Tablespace	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. 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.	SCHEMA_E

Type	Parameter	Description	Example Value
	Table Name	Name of the table 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 the table or directly enter a table name. If the desired table is not displayed, check whether the table exists or whether the login account has the permission to query metadata. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*). **NOTE** The table name can contain wildcard characters (*), which is used to export all tables whose names start with a certain prefix or end with a certain suffix. The number and types of fields in the tables must be the same. The examples are as follows: • **table*** indicates that all tables whose names starting with **table** are exported. • ***table** indicates that all tables whose names ending with **table** are exported. • ***table*** indicates that all tables whose names containing **table** are exported.	table

Type	Parameter	Description	Example Value
Advanced attributes	Partition Column	This parameter is displayed when **Use SQL Statement** is set to **No**, indicating that a field used to split data during data extraction. CDM splits a job into multiple tasks based on this field and executes the tasks concurrently. Fields with data distributed evenly are used, such as the sequential number field. Click the icon next to the text box to go to the page for selecting a field or directly enter a field. **NOTE** The following types of partition columns are supported: TINYINT, SMALLINT, INTEGER, BIGINT, REAL, FLOAT, DOUBLE, NUMERIC, DECIMAL, BIT, BOOLEAN, DATE, TIME, and TIMESTAMP. It is recommended that the partition column have an index.	id
	WHERE Clause	WHERE clause used to specify the data extraction range. This parameter is displayed when **Use SQL Statement** is set to **No**. If this parameter is not set, the entire table is extracted. You can set a date macro variable to extract data generated on a specific date. For details, see **Incremental Migration of Relational Databases**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	DS='${dateformat(yyyy-MM-dd,-1,DAY)}'
	Null in Partition Column	Whether the partition column can contain null values	Yes

5.4 Destination Job Parameters

5.4.1 To OBS

If the destination link of a job is an **OBS link**, that is, data is to be imported to OBS, configure the destination job parameters based on **Table 5-29**.

Advanced attributes are optional and not displayed by default. You can click **Show Advanced Attributes** to display them.

Table 5-29 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Bucket Name	Name of the OBS bucket that data will be written to	bucket_2
	Write Directory	OBS directory to which data will be written. Do not add **/** in front of the directory name. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	directory/
	File Format	Format in which data is written. The options are as follows: • **CSV**: Data is written in CSV format, which is used for migrating data tables to files. • **Binary**: Files will be transferred directly. CDM writes the files without changing their format. This setting is suitable for file migration. If data is migrated between file-related data sources, such as FTP, SFTP, OBS, and HDFS, the value of **File Format** must the same as the source file format. **NOTE** • The format can only be CSV when the source link is an MRS Hive link. • If the source is an FTP/SFTP server, only the binary format is supported.	CSV

Category	Parameter	Description	Example Value
	Duplicate File Processing Method	This parameter is available when the migration source is HDFS. Files with the same name and size are identified as duplicate files. If there are duplicate files during data writing, the following methods are available: • Replace • Skip • Stop job For details, see **Incremental File Migration**.	Skip
Advanced attributes	Encryption	Whether to encrypt the uploaded data and the encryption mode. The options are as follows: • **None**: Data is written without encryption. • **KMS**: KMS in Data Encryption Workshop (DEW) is used for encryption. If KMS encryption is enabled, MD5 verification for data cannot be performed. For details, see **Encryption and Decryption During File Migration**.	KMS
	KMS ID	Data encryption key. This parameter is displayed when **Encryption** is set to **KMS**. Click next to the text box to select the KMS key that was created in DEW. • If the KMS key of the same project as that of the CDM cluster is used, you do not need to modify **Project ID**. • If the KMS key of another project is used, you need to modify **Project ID**.	53440ccb-3e73-4700-98b5-71ff5476e621
	Project ID	ID of the project to which KMS ID belongs. The default value is the ID of the project to which the current CDM cluster belongs. • If KMS and the CDM cluster are in the same project, retain the default value of **Project ID**. • If KMS of another project is used, set this parameter to the ID of the project to which KMS belongs.	9bd7c4bd54e5417198f9591bef07ae67

Category	Parameter	Description	Example Value
	Copy Content-Type	This parameter is displayed only when **File Format** is **Binary**, and both the migration source and destination are object storage. If you set this parameter to **Yes**, the Content-Type attribute of the source file is copied during object file migration. This function is mainly used for static website migration. The Content-Type attribute cannot be written to Archive buckets. Therefore, if you set this parameter to **Yes**, the migration destination must be a non-Archive bucket.	No
	Line Separator	Lind feed character in a file. By default, the system automatically identifies **\n**, **\r**, and **\r\n**. This parameter is not used when **File Format** is set to **Binary**.	\n
	Field Delimiter	Field delimiter in the file. This parameter is not used when **File Format** is set to **Binary**.	,
	File Size	This parameter is displayed only when the migration source is a database. Files are partitioned as multiple files by size so that they can be exported in proper size. The unit is MB.	1024
	Validate MD5 Value	The MD5 value can be verified only when files are transferred in **Binary** format. KMS encryption cannot be used if the MD5 value needs to be verified. Calculate the MD5 value of the source files and verify it with the MD5 value returned by OBS. If an MD5 file exists on the migration source, the system directly reads the MD5 file from the migration source and verifies it with the MD5 value returned by OBS. For details, see **MD5 Verification**.	Yes
	Record MD5 Verification Result	Whether to record the MD5 verification result when **Validate MD5 Value** is set to **Yes**	Yes
	Record MD5 Link	OBS link to which the MD5 verification result will be written	obslink

Category	Parameter	Description	Example Value
	Record MD5 Bucket	OBS bucket to which the MD5 verification result will be written	cdm05
	Record MD5 Directory	Directory to which the MD5 verification result will be written	/md5/
	Encoding Type	Encoding type, for example, **UTF-8** or **GBK**. This parameter is not used when **File Format** is set to **Binary**.	GBK
	Use Quote Character	This parameter is displayed only when **File Format** is **CSV**. It is used when database tables are migrated to file systems. If you set this parameter to **Yes** and a field in the source data table contains a field delimiter or line separator, CDM uses double quotation marks (") as the quote character to quote the field content as a whole to prevent a field delimiter from dividing a field into two fields, or a line separator from dividing a field into different lines. For example, if the **hello,world** field in the database is quoted, it will be exported to the CSV file as a whole.	No
	Use First Row as Header	This parameter is displayed only when data is exported from a relational database to OBS and **File Format** is set to **CSV**. When a table is migrated to a CSV file, CDM does not migrate the heading line of the table by default. If you set this parameter to **Yes**, CDM writes the heading line of the table to the file.	No
	Job Success Marker File	Whether to generate a marker file with a custom name in the destination directory after a job is executed successfully. If you do not specify a file name, this function is disabled by default.	finish.txt

Category	Parameter	Description	Example Value
	Folder Mode	This parameter is available only when data is exported from a relational database to OBS. If this function is enabled, generated files are named in the following format: *Root directory-Table name-Data type-Data folder format*. Example: **raw_schema/tbl_student/datas/tbl_student_1.csv**	Yes
	Blog/Clog File Name Extension	This parameter is available only when **Folder Mode** is set to **Yes**. It specifies the extension for the names of the files that contain custom Blob/Clog data in folder mode.	.dat/.jpg/.png
	Customize Hierarchical Directory	If this parameter is set to **Yes**, the files after migration can be stored in a custom directory. That is, only files are migrated. The directories to which the files belong are not migrated.	Yes
	Hierarchical Directory	Custom storage directory for files after migration. The time macro variable is supported. **NOTE** If the source link is a relational database link, the directory name consists of the source table name and a custom directory name. In other scenarios, the directory is a custom directory.	${dateformat(yyyy-MM-dd HH:mm:ss, -1, DAY)}

Category	Parameter	Description	Example Value
	Customize File Name	This parameter is displayed only when data is exported from a relational database to OBS and **File Format** is set to **CSV**. This parameter specifies the name of the file generated by OBS. The options are as follows: • **Character string**: Special characters are allowed. For example, if this parameter is set to **cdm#**, the name of the generated file is **cdm#.csv**. • Macro variable of time: If this parameter is set to **${timestamp()}**, the name of the generated file is **1554108737.csv**. • Macro variable of table name: If this parameter is set to **${tableName}**, the name of the generated file is the source table name **sqltabname.csv**. • Macro variable of version number: If this parameter is set to **${version}**, the name of the generated file is the cluster version number **2.9.2.200.csv**. • Any combination of the character string and macro variable (macro variable of time, table name, or version number). For example, if this parameter is set to **cdm#${timestamp()}_${version}**, the name of the generated file is **cdm#1554108737_2.9.2.200.csv**.	cdm

5.4.2 To HDFS

If the destination link of a job is an **HDFS link**, configure the destination job parameters based on **Table 5-30**.

Table 5-30 Parameter description

Parameter	Description	Example Value
Write Directory	HDFS directory to which data will be written. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	/user/output
File Format	Format in which data is written. The options are as follows: • **CSV**: Data is written in CSV format, which is used for migrating data tables to files. • **Binary**: Files will be transferred directly. CDM writes the files without changing their format. This setting is suitable for file migration. If data is migrated between file-related data sources, such as FTP, SFTP, OBS, and HDFS, the value of **File Format** must the same as the source file format.	CSV
Duplicate File Processing Method	This parameter is available when the migration source is a file data source, such as HTTP, FTP, SFTP, OBS, and HDFS. Files with the same name and size are identified as duplicate files. If there are duplicate files during data writing, the following methods are available: • Replace • Skip • Stop job	Stop job

Parameter	Description	Example Value
Compression Format	File compression format after data writing. The following compression formats are supported: • **None**: The files are not compressed. • **DEFLATE**: The files are compressed in DEFLATE format. • **gzip**: The files are compressed in gzip format. • **bzip2**: The files are compressed in bzip2 format. • **LZ4**: The files are compressed in LZ4 format. • **Snappy**: The files are compressed in snappy format.	Snappy
Line Separator	Lind feed character in a file. By default, the system automatically identifies **\n**, **\r**, and **\r\n**. This parameter is not used when **File Format** is set to **Binary**.	\n
Field Delimiter	Field delimiter in the file. This parameter is not used when **File Format** is set to **Binary**.	,
Use Quote Character	This parameter is displayed only when **File Format** is **CSV**. It is used when database tables are migrated to file systems. If you set this parameter to **Yes** and a field in the source data table contains a field delimiter or line separator, CDM uses double quotation marks (") as the quote character to quote the field content as a whole to prevent a field delimiter from dividing a field into two fields, or a line separator from dividing a field into different lines. For example, if the **hello,world** field in the database is quoted, it will be exported to the CSV file as a whole.	No
Use First Row as Header	When a table is migrated to a CSV file, CDM does not migrate the heading line of the table by default. If you set this parameter to **Yes**, CDM writes the heading line of the table to the file.	No
Write to Temporary File	Whether to write the binary file to a **.tmp** file first. After the migration is successful, run the **rename** or **move** command at the migration destination to restore the file.	No
Job Success Marker File	Whether to generate a marker file with a custom name in the destination directory after a job is executed successfully. If you do not specify a file name, this function is disabled by default.	finish.txt

Parameter	Description	Example Value
Customize Hierarchical Directory	Users can customize the directory hierarchy of files. Example: [Table name]/[Year]/[Month]/[Day]/[Data file name]. csv	-
Hierarchical Directory	Used to specify the directory level of a file, with time macro supported (the time format is yyyy/MM/dd). If this parameter is left blank, the directory does not have a hierarchical structure. **NOTE** If the source link is a relational database link, the directory name consists of the source table name and a custom directory name. In other scenarios, the directory is a custom directory.	${dateformat(yyyy/MM/dd, -1, DAY)}
Encryption	This parameter is displayed only when **File Format** is set to **Binary**. Whether to encrypt the uploaded data. The options are as follows: • **None**: Data is written without encryption. • **AES-256-GCM**: The AES 256-bit encryption algorithm is used to encrypt data. Currently, only the AES-256-GCM (NoPadding) encryption algorithm is supported. This parameter is used for encryption at the migration destination and decryption at the migration source. For details, see **Encryption and Decryption During File Migration**.	AES-256-GCM
DEK	This parameter is displayed only when **Encryption** is set to **AES-256-GCM**. The key consists of 64 hexadecimal numbers. Remember the key configured here because the decryption key must be the same as that configured here. If the encryption and decryption keys are inconsistent, the system does not report an exception, but the decrypted data is incorrect.	DD0AE00DFECD78BF051BCFDA25BD4E320DB0A7AC75A1F3FC3D3C56A457DCDC1B
IV	This parameter is displayed only when **Encryption** is set to **AES-256-GCM**. The initialization vector consists of 32 hexadecimal numbers. Remember the initialization vector configured here because the initialization vector used for decryption must be the same as that configured here. If the encryption and decryption keys are inconsistent, the system does not report an exception, but the decrypted data is incorrect.	5C91687BA886EDCD12ACBC3FF19A3C3F

NOTE

HDFS supports the **UTF-8** encoding only. Retain the default value **UTF-8**.

5.4.3 To HBase/CloudTable

If the destination link of a job is an **HBase link** or **CloudTable link**, configure the destination job parameters based on **Table 5-31**.

Table 5-31 Parameter description

Parameter	Description	Example Value
Table Name	Name of the HBase table to which data will be written. If you want to create an HBase table, you can copy the field names from the migration source. Click the icon next to the text box. The dialog box for selecting the table is displayed. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	TBL_2
Clear Data Before Import	Whether the data in the destination table is cleared before data import. The options are as follows: • **Yes**: The data is cleared. • **No**: The data is not cleared. Instead, it will be added to the existing table.	Yes

Parameter	Description	Example Value
Auto Table Creation	This parameter is displayed only when the source is a relational database. The options are as follows: ● **Non-auto creation**: CDM will not automatically create a table. ● **Auto creation**: If the destination database does not contain the table specified by **Table Name**, CDM will automatically create the table. If the table specified by **Table Name** already exists, no table is created and data is written to the existing table. **NOTE** The automatically created HBase table contains the column family and coprocessor information. For other attributes, default values are retained.	Non-auto creation
Rowkey Delimiter	(Optional) Used to combine multiple columns as a rowkey. Spaces are used by default.	,
Rowkey Data Redundancy	(Optional) Whether to write the rowkey data into HBase columns. The default value is **No**.	No
Compression Format	(Optional) Compression format used in creating an HBase table. The default value is **None**. ● **None**: The files are not compressed. ● **Snappy**: The files are compressed in snappy format. ● **gzip**: The files are compressed in gzip format.	None
Write WAL	Whether to enable Write Ahead Log (WAL) of HBase. The options are as follows: ● **Yes**: If the HBase server breaks down after the function is enabled, you can replay the operations that have not been performed in WAL. ● **No**: If you set this parameter to **No**, the write performance is improved. However, if the HBase server breaks down, data may be lost.	No

Parameter	Description	Example Value
Match Data Type	• **Yes**: Data of the Short, Int, Long, Float, Double, and Decimal columns in the source database is converted into Byte[] arrays (binary) and written into HBase. Other types of data are written as character strings. If several types of data mentioned above are combined as rowkeys, they will be written as character strings. This function saves storage space. In specific scenarios, the rowkey distribution is evener. • **No**: All types of data in the source database are written into HBase as character strings.	No

5.4.4 To Hive

If the destination link of a job is a **Hive link**, configure the destination job parameters based on **Table 5-32**.

Table 5-32 Parameter description

Parameter	Description	Example Value
Database Name	Database name. Click the icon next to the text box. The dialog box for selecting the database is displayed.	default
Table Name	Destination table name. Click the icon next to the text box. The dialog box for selecting the table is displayed. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	TBL_X

Parameter	Description	Example Value
Auto Table Creation	This parameter is displayed only when the source is a relational database. The options are as follows: ● **Non-auto creation**: CDM will not automatically create a table. ● **Auto creation**: If the destination database does not contain the table specified by **Table Name**, CDM will automatically create the table. If the table specified by **Table Name** already exists, no table is created and data is written to the existing table. ● **Deletion before creation**: CDM deletes the table specified by **Table Name**, and then creates the table again. **NOTE** ● Only column comments are synchronized during automatic table creation. Table comments are not synchronized. ● Primary keys cannot be synchronized during automatic table creation.	Non-auto creation
Clear Data Before Import	Whether the data in the destination table is cleared before data import. The options are as follows: ● **Yes**: The data is cleared. ● **No**: The data is not cleared. Instead, it will be added to the existing table.	Yes
Partition to Clear	This parameter is available when **Clear Data Before Import** is set to **Yes**. When you enter the information about the partitions to be cleared, the data in the partitions will be cleared.	Single partition: **year=2020,location=sun** Multiple partitions: **['year=2020,location=sun', 'year=2021,location=earth']**

Parameter	Description	Example Value
Executing Analyze Statements	After all data is written, the ANALYZE TABLE statement is asynchronously executed to accelerate the Hive table query. The SQL statement is as follows: • Non-partitioned table: **ANALYZE TABLE tablename COMPUTE STATISTICS** • Partitioned table: **ANALYZE TABLE tablename PARTITION(partcol1[=val1], partcol2[=val2], ...) COMPUTE STATISTICS** **NOTE** Parameter **Executing Analyze Statements** applies only to the migration of a single table.	Yes

NOTE

- When Hive serves as the destination end, a table whose storage format is ORC is automatically created.
- Due to file format restrictions, complex data can be written only in ORC or Parquet format.
- If the source Hive contains both the array and map types of data, the destination table format can only be the ORC or parquet complex type. If the destination table format is RC or TEXT, the source data will be processed and can be successfully written.
- As the map type is an unordered data structure, the data type may change after a migration.
- If Hive serves as the migration destination and the storage format is Textfile, delimiters must be explicitly specified in the statement for creating Hive tables. The following is an example:

```
CREATE TABLE csv_tbl(
smallint_value smallint,
tinyint_value tinyint,
int_value int,
bigint_value bigint,
float_value float,
double_value double,
decimal_value decimal(9, 7),
timestmamp_value timestamp,
date_value date,
varchar_value varchar(100),
string_value string,
char_value char(20),
boolean_value boolean,
binary_value binary,
varchar_null varchar(100),
string_null string,
char_null char(20),
int_null int
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
"separatorChar" = "\t",
"quoteChar"     = "'",
"escapeChar"    = "\\"
)
STORED AS TEXTFILE;
```


5.4.5 To MySQL/SQL Server/PostgreSQL

Table 5-33 lists the destination job parameters when the destination link is an MySQL, SQL Server, or PostgreSQL link.

Table 5-33 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Schema/ Tablespace	Name of the database to which data will be written. The schema can be automatically created. Click the icon next to the text box to select a schema or tablespace.	schema
	Auto Table Creation	This parameter is displayed only when the source is a relational database. The options are as follows: • **Non-auto creation**: CDM will not automatically create a table. • **Auto creation**: If the destination database does not contain the table specified by **Table Name**, CDM will automatically create the table. If the table specified by **Table Name** already exists, no table is created and data is written to the existing table. • **Deletion before creation**: CDM deletes the table specified by **Table Name**, and then creates the table again.	Non-auto creation
	Table Name	Name of the table to which data will be written. Click the icon next to the text box. The dialog box for selecting the table is displayed. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	table

Category	Parameter	Description	Example Value
	Clear Data Before Import	Whether to clear the data in the destination table before data import. The options are as follows: • **Do not clear**: The data in the destination table is not cleared before data import. The imported data is just added to the table. • **Clear all data**: All data is cleared from the destination table before data import. • **Clear part of data**: Part of the data in the destination table is cleared before data import. If you select **Clear part of data**, you must configure **WHERE Clause** to specify which part will be deleted.	Clear part of data
	WHERE Clause	If **Clear Data Before Import** is set to **Clear part of data**, data in the destination table will be deleted based on the WHERE clause after the configuration is complete and before the import starts.	age > 18 and age <= 60
	Constraint Conflict Handling	How to handle data conflicts when data is being imported to RDS for MySQL • **insert into**: When a primary key or unique index conflict occurs, data cannot be written and will become dirty data. • **replace into**: When a primary key or unique index conflict occurs, the original row is deleted and a new row is inserted to replace all the fields in the original row. • **on duplicate key update**: When a primary key or unique index conflict occurs in a row in the destination table, the data columns except the unique constraint column in this row are updated.	insert into

Category	Parameter	Description	Example Value
Advanced parameters	Import to Staging Table	If you set this parameter to **Yes**, the transaction mode is enabled. CDM automatically creates a temporary table and imports data to the temporary table. After the data is imported successfully, it is migrated to the destination table in transaction mode. If the import fails, the destination table is rolled back to the state before the job starts. For details, see **Migration in Transaction Mode**. The default value is **No**, indicating that CDM directly imports the data to the destination table. In this case, if the job fails to be executed, the data that has been imported to the destination table will not be rolled back automatically. **NOTE** If you select **Clear part of data** or **Clear all data** for **Clear Data Before Import**, CDM does not roll back the deleted data in transaction mode.	No
	Extend Field Length	When **Auto creation** is selected, the length of the character fields can be extended to three times the original length and then written to the destination table. If the encoding types of the source and destination databases are different, but the character fields in the source and destination tables are the same, errors may occur during data migration due to character length difference. **NOTE** When this function is enabled, some fields consume three times the storage space of the user.	No
	Use NOT NULL Constraint	If you choose to create a target table automatically and specify the NOT NULL constraint, keep the NOT NULL constraints of the source and target tables consistent.	Yes
	Prepare for Data Import	The SQL statement that is first executed before a task is executed. Currently, only one SQL statement can be executed in wizard mode.	create temp table
	Complete Statement After Data Import	The SQL statement that is executed after a task is executed. Currently, only one SQL statement can be executed.	merge into

Category	Parameter	Description	Example Value
	Loader Threads	Number of threads started in each loader. A larger number allows more concurrent write operations. **NOTE** This parameter is unavailable if **Constraint Conflict Handling** is set to **replace into** or **on duplicate key update**.	1

5.4.6 To Oracle

If the destination link of a job is an **Oracle database link**, configure the destination job parameters based on **Table 5-34**.

Table 5-34 Parameter description

Type	Parameter	Description	Example Value
Basic parameters	Schema/ Tablespace	Name of the database to which data will be written. The schema can be automatically created. Click the icon next to the text box to select a schema or tablespace.	schema
	Table Name	Name of the table to which data will be written. Click the icon next to the text box. The dialog box for selecting the table is displayed. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	table

Type	Parameter	Description	Example Value
	Clear Data Before Import	Whether to clear the data in the destination table before data import. The options are as follows: • **Do not clear**: The data in the destination table is not cleared before data import. The imported data is just added to the table. • **Clear all data**: All data is cleared from the destination table before data import. • **Clear part of data**: Part of the data in the destination table is cleared before data import. If you select **Clear part of data**, you must configure **WHERE Clause** to specify which part will be deleted.	Clear part of data
	WHERE Clause	If **Clear Data Before Import** is set to **Clear part of data**, data in the destination table will be deleted based on the WHERE clause after the configuration is complete and before the import starts.	age > 18 and age <= 60
Advanced parameters	Import to Staging Table	If you set this parameter to **Yes**, the transaction mode is enabled. CDM automatically creates a temporary table and imports data to the temporary table. After the data is imported successfully, it is migrated to the destination table in transaction mode. If the import fails, the destination table is rolled back to the state before the job starts. For details, see **Migration in Transaction Mode**. The default value is **No**, indicating that CDM directly imports the data to the destination table. In this case, if the job fails to be executed, the data that has been imported to the destination table will not be rolled back automatically. **NOTE** If you select **Clear part of data** or **Clear all data** for **Clear Data Before Import**, CDM does not roll back the deleted data in transaction mode.	No
	Prepare for Data Import	The SQL statement that is first executed before a task is executed. Currently, only one SQL statement can be executed in wizard mode.	create temp table

Type	Parameter	Description	Example Value
	Complete Statement After Data Import	The SQL statement that is executed after a task is executed. Currently, only one SQL statement can be executed.	merge into
	Loader Threads	Number of threads started in each loader. A larger number allows more concurrent write operations. **NOTE** This parameter is unavailable if **Constraint Conflict Handling** is set to **replace into** or **on duplicate key update**.	1

5.4.7 To DWS

If the destination link of a job is a **DWS link**, configure the destination job parameters based on **Table 5-35**.

Table 5-35 Parameter description

Parameter	Description	Example Value
Schema / Tablespace	Name of the database to which data will be written. The schema can be automatically created. Click the icon next to the text box to select a schema or tablespace.	schema

Parameter	Description	Example Value
Auto Table Creation	This parameter is displayed only when the source is a relational database. The options are as follows: • **Non-auto creation**: CDM will not automatically create a table. • **Auto creation**: If the destination database does not contain the table specified by **Table Name**, CDM will automatically create the table. If the table specified by **Table Name** already exists, no table is created and data is written to the existing table. • **Deletion before creation**: CDM deletes the table specified by **Table Name**, and then creates the table again. **Field Mapping in Automatic Table Creation on DWS** describes the field mapping between the DWS tables created by CDM and source tables. **NOTE** Only column comments are synchronized during automatic table creation. Table comments are not synchronized.	Non-auto creation
Table Name	Name of the table to which data will be written. Click the icon next to the text box. The dialog box for selecting the table is displayed. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	table
Compress Data	Whether to compress data when data is imported to DWS and **Auto creation** is selected	No

Parameter	Description	Example Value
Storage Mode	When data is imported to DWS and **Auto Creation** is selected, you can specify the data storage mode: • **Row-based**: Row-based storage. It is used for point queries (index-based simple queries with fewer return records), or the scenario that requires a large number of addition, deletion, and modification operations. • **Column-based**: Column-based storage. It is used for statistical analysis queries (group and join scenarios) or ad hoc queries (query conditions are uncertain and indexes can hardly be used to scan row-based tables).	Row-based
Import Mode	Mode for importing data to DWS • In COPY mode, the source data is copied to the DataNode of DWS after passing through the management node. • In UPSERT mode, if a primary key or unique constraint conflict occurs, other data columns, except the primary key and unique constraint column, are updated.	COPY
Clear Data Before Import	Whether to clear the data in the destination table before data import. The options are as follows: • **Do not clear**: The data in the destination table is not cleared before data import. The imported data is just added to the table. • **Clear all data**: All data is cleared from the destination table before data import. • **Clear part of data**: Part of the data in the destination table is cleared before data import. If you select **Clear part of data**, you must configure **WHERE Clause** to specify which part will be deleted.	Clear part of data
WHERE Clause	If **Clear Data Before Import** is set to **Clear part of data**, data in the destination table will be deleted based on the WHERE clause after the configuration is complete and before the import starts.	age > 18 and age <= 60

Parameter	Description	Example Value
Import to Staging Table	If you set this parameter to **Yes**, the transaction mode is enabled. CDM automatically creates a temporary table and imports data to the temporary table. After the data is imported successfully, it is migrated to the destination table in transaction mode. If the import fails, the destination table is rolled back to the state before the job starts. . The default value is **No**, indicating that CDM directly imports the data to the destination table. In this case, if the job fails to be executed, the data that has been imported to the destination table will not be rolled back automatically. **NOTE** If you select **Clear part of data** or **Clear all data** for **Clear Data Before Import**, CDM does not roll back the deleted data in transaction mode.	No
Extending field length	When **Auto creation** is selected, the length of the character fields can be extended to three times the original length and then written to the destination table. If the encoding types of the source and destination databases are different, but the character fields in the source and destination tables are the same, errors may occur during data migration due to character length difference. When a character field containing Chinese characters is imported to DWS, the length of the character field must be automatically increased by three times. If a job fails to be executed and an error message similar to **value too long for type character varying** exists in the log when you import Chinese characters to DWS, you can enable this function to solve the problem. **NOTE** When this function is enabled, some fields consume three times the storage space of the user.	No
Use NOT NULL Constraint	If you choose to create a target table automatically and specify the NOT NULL constraint, keep the NOT NULL constraints of the source and target tables consistent.	Yes
Prepare for Data Import	The SQL statement that is first executed before a task is executed. Currently, only one SQL statement can be executed in wizard mode.	create temp table

Parameter	Description	Example Value
Complete Statement After Data Import	The SQL statement that is executed after a task is executed. Currently, only one SQL statement can be executed.	merge into
Loader Threads	Number of threads started in each loader. A larger number allows more concurrent write operations.	1

Field Mapping in Automatic Table Creation on DWS

Figure 5-8 describes the field mapping between DWS tables created by CDM and source tables. For example, if you use CDM to migrate the Oracle database to DWS, CDM automatically creates a table on DWS and maps the **NUMBER(3,0)** field of the Oracle database to the **SMALLINT** field of DWS.

Figure 5-8 Field mapping in automatic table creation

Source Database Type							Destination Database Type
Oracle	MySQL	SQL Server	PostgreSQL	Db2	GaussDB	SAP HANA	DWS
NUMBER(p,0) (p=3 or p=5)	SMALLINT,TINYINT	SMALLINT,TINYINT	SMALLINT	DECIMAL	SMALLINT	SMALLINT,TINYINT	SMALLINT
NUMBER(10,0)	INT	INT	INTEGER	INT	INTEGER	INTEGER	INTEGER
NUMBER(19,0)	BIGINT	BIGINT	BIGINT	DECIMAL	BIGINT	BIGINT	BIGINT
None	None	None	OID	None	OID	CHAR(128)	OID
NUMBER(p,s) (0 < p <= 38)	DECIMAL(p,s) (0 < p <= 65)	DECIMAL(p,s) (0 < p <= 30)	NUMERIC(p,s) (p <= 1000)	DECIMAL	NUMERIC(p,s) (p <= 1000)	DECIMAL(p,s) (0 < p <= 38)	NUMERIC(p,s) (p <= 1000)
RAW	BINARY	BINARY	BYTEA	BINARY	BYTEA	BINARY	BYTEA
CHAR	CHAR	CHAR	CHAR	CHAR	CHAR	CHAR(p) (p <= 2000)	CHAR
NCHAR	NCHAR	NCHAR	NCHAR	NCHAR	NCHAR	NCHAR(p) (p <= 5000)	NCHAR
DATE	DATE	DATE	DATE	DATE	DATE	DATE	DATE
DATE	DATETIME	DATETIME2	TIMESTAMP	TIMESTAMP	TIMESTAMP	TIMESTAMP	TIMESTAMP
VARCHAR2(p) (p <= 4000)	VARCHAR	VARCHAR(p) (if p >= 8000 p=max)	VARCHAR(p) (p <= 10485760)	VARCHAR	VARCHAR(p) (p <= 10485760)	VARCHAR(p) (p <= 5000)	VARCHAR(p) (p <= 10485760)
FLOAT	DOUBLE	FLOAT	DOUBLE PRECISION	FLOAT	DOUBLE PRECISION	DOUBLE	DOUBLE PRECISION
FLOAT	REAL	FLOAT	REAL	FLOAT	REAL	REAL	REAL
CLOB	TEXT	TEXT	TEXT	TEXT	TEXT	CLOB	TEXT
DATE	None	TIME	TIME	TIME	TIME	TIME	TIME
BOOLEAN	None	None	BOOLEAN	BOOLEAN	BOOLEAN	BOOLEAN	BOOLEAN

NOTE

Indexes cannot be created in automatic table creation scenarios.

5.4.8 To DDS

If the destination link of a job is a **DDS link**, configure the destination job parameters based on **Table 5-36**.

Table 5-36 Parameter description

Parameter	Description	Example Value
Database Name	Database to which data is to be imported	ddsdb
Collection Name	Collection of data to be imported, which is similar to the table name of a relational database. Click the icon next to the text box to go to the page for selecting the table or directly enter a table name. If the desired table is not displayed, check whether the table exists or whether the login account has the permission to query metadata.	COLLECTION

5.4.9 To Elasticsearch/CSS

If the destination link of a job is a link described in **Link to Elasticsearch** or **Link to CSS**, configure the destination job parameters based on **Table 5-37**.

Table 5-37 Job parameters when Elasticsearch/CSS is the destination

Parameter	Description	Example Value
Index	Elasticsearch index, which is similar to the name of a relational database. CDM supports automatic creation of indexes and field types. The index and field type names can contain only lowercase letters.	index
Type	Elasticsearch type, which is similar to the table name of a relational database. The type name can contain only lowercase letters. **NOTE** Elasticsearch 7.x and later versions do not support custom types. Instead, only the **_doc** type can be used. In this case, this parameter does not take effect even if it is set.	type

Parameter	Description	Example Value
Pipeline ID	Pipeline used to convert the data format after data is transferred to Elasticsearch. Pipeline IDs are ready for use after being created in Kibana.	pipeline_id
Write ES with Routing	If you enable this function, a column can be written to Elasticsearch as a route. **NOTE** Before enabling this function, create indexes at the destination to improve the query efficiency.	No
Route Column	This parameter is available when **Write ES with Routing** is set to **Yes**. It specifies the destination routing column. If the destination index exists but the column information cannot be obtained, you can manually enter the column. The route column can be empty. If it is empty, no routing value is specified for the data written to Elasticsearch.	value1
Periodically Create Index	For streaming jobs that continuously write data to Elasticsearch, CDM periodically creates indexes and writes data to the indexes, which helps you delete expired data. The indexes can be created based on the following periods: • **Every hour**: CDM creates indexes on the hour. The new indexes are named in the format of *Index name+Year+Month+Day+Hour*, for example, **index2018121709**. • **Every day**: CDM creates indexes at 00:00 every day. The new indexes are named in the format of *Index name+Year+Month+Day*, for example, **index20181217**. • **Every week**: CDM creates indexes at 00:00 every Monday. The new indexes are named in the format of *Index name+Year+Week*, for example, **index201842**. • **Every month**: CDM creates indexes at 00:00 on the first day of each month. The new indexes are named in the format of *Index name+Year +Month*, for example, **index201812**. • **Do not create**: Do not create indexes periodically. When extracting data from a file, you must configure a single extractor, which means setting **Concurrent Extractors** to **1**. Otherwise, this parameter is invalid.	Every hour

5.4.10 To DLI

If the destination link of a job is a **DLI link**, configure the destination job parameters based on **Table 5-38**.

NOTE

When you use CDM to migrate data to DLI, DLI generates data files in the *dli-trans** temporary OBS bucket. Therefore, you need to grant the account corresponding to the AK/SK the permissions to read and write the *dli-trans** bucket and create directories. For details about how to add OBS permission policies, see **Adding an OBS Bucket Policy**.

Table 5-38 Parameter description

Parameter	Description	Example Value
Resource Queue	Resource queue to which the destination table belongs The default queue of DLI cannot be used for migration jobs. You need to create a SQL queue in DLI. For details about how to create a queue, see **Creating a Queue**.	cdm
Database Name	Name of the database to which data will be written	dli
Table Name	Name of the table to which data will be written	car_detail
Clear Data Before Import	Whether to clear data in the destination table before data import If this parameter is set to **Yes**, data in the destination table will be cleared before the task is started.	No
Convert empty strings to null	If this parameter is set to **Yes**, an empty string is regarded as null.	No
Data Clearing Mode	This parameter is available when **Clear Data Before Import** is set to **Yes**. **TRUNCATE**: deletes standard data. **INSERT_OVERWRITE**: overwrites existing data with inserted data. **NOTE** If the source link is a Kafka link and **Clear Data Before Import** is set to **Yes**, **INSERT_OVERWRITE** is unavailable.	TRUNCATE

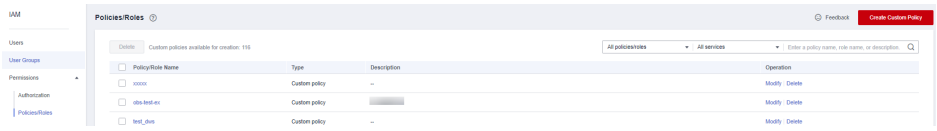
Parameter	Description	Example Value
Partition	This parameter is available when **Clear Data Before Import** is set to **Yes**. When you enter partitions, data in these partitions will be cleared.	year=2020,location=sun

Adding an OBS Bucket Policy

Step 1 Log in to the IAM console.

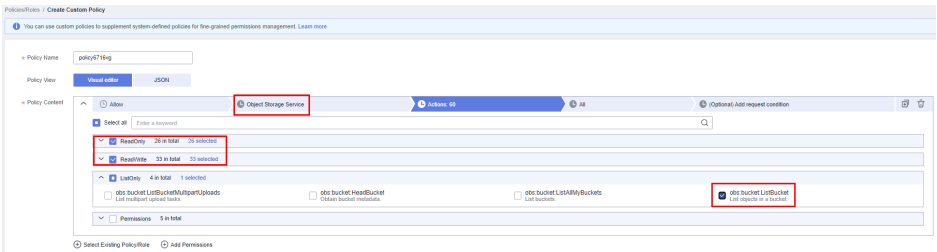
Step 2 In the navigation pane, choose **Permissions** > **Policies/Roles** and click **Create Custom Policy** in the upper right corner.

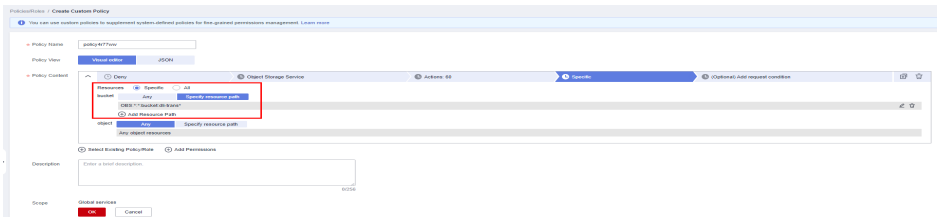
Figure 5-9 Creating a custom policy



Step 3 Enter a policy name and set **Policy Content**.

Figure 5-10 Configuring the policy





Step 4 Enter the policy description and click **OK**.

----End

5.4.11 To MRS Hudi

If the destination link of a job is an **MRS Hudi link**, configure the destination job parameters based on **Table 5-39**.

Table 5-39 Parameter description

General Configuration		
Item	Configuration Description	Recommended Configuration
Destination Link Name	MRS Hudi link	hudi_to_cdm
Database Name	Database name. Click the icon next to the text box. The dialog box for selecting the database is displayed.	dbadmin
Table Name	Click the icon next to the text box. The dialog box for selecting the table is displayed. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. You can use macro variables of date and time in a scheduled job to synchronize incremental data periodically. For details, see **Using Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).	cdm
Auto Table Creation	Whether to automatically create Hudi tables • **Non-auto creation**: CDM will not automatically create a table. • **Auto creation**: If the destination database does not contain the table specified by **Table Name**, CDM will automatically create the table. If the table specified by **Table Name** already exists, no table is created and data is written to the existing table.	Non-auto creation
Clear Data Before Import	Whether the data in the destination table is cleared before data import. The options are as follows: • **Yes**: The data is cleared. • **No**: The data is not cleared. Instead, it will be added to the existing table.	No

General Configuration		
Full Data Mode to Write Hoodie	Hoodie write mode. The default value is **Yes**, indicating the full mode. Value **No** indicates the microbatch mode. • In full mode, data is asynchronously written to Hoodie by fragments, which is suitable for writing all data at a time. • In microbatch mode, data is asynchronously written to Hoodie in batches. This mode is suitable if there are strict SLA requirements on the import time, a small number of resources are required, or the MOR table storage types are compressed online. **NOTE** This mode cannot be changed during a retry upon failure.	Yes
Batch Size	This parameter is available when **Full Data Mode to Write Hoodie** is set to **No**. It specifies the number of data rows written to Hoodie in a single batch. The default value is **100000**.	100000
Use the import time field	A field marked as the import time field. If a table is automatically created, this field is automatically added to the table creation statement. When data is written to Hudi, the value of this field is replaced by the current time. If the table is not automatically created, select the existing import time field.	Yes
Data import time field name	This parameter is available when **Use the import time field** is set to **Yes**. It specifies the time when data is written to Hudi. **NOTE** • If the destination table already has an import time field, you can directly use the existing timestamp field. • In the automatic table creation scenario, this field is concatenated to the table creation statement and it is a timestamp. The field name cannot be the same as that of any source field (including custom fields).	cdc_last_update_date
Hudi Table Creation Configuration		

General Configuration		
Location	OBS or HDFS path where database table files are stored	-
Hudi Table Type	Storage type of the Hudi table • **MOR**: Data is written to a log file in avro format and then merged into a Parquet file when being read. • **COW**: Data is directly written to a Parquet file.	MOR
Hudi table primary key	Primary keys for creating a Hudi table. Use commas (,) to separate multiple keys.	-
Hudi Table Key Generator Class	Primary key generation type, which implements **org.apache.hudi.keygen.KeyGenerator** to extract key values from input records.	-
Hudi table pre-combine key	If two records have the same primary key, the record with a larger **precombine** value is retained. **NOTE** If no time field is available, you can set a field that is the same as the primary key. When a primary key conflict occurs, the latest record is retained.	ts
Hudi Table Partition Fields	Partition fields for creating a Hudi table. Use commas (,) to separate multiple fields.	-
Hudi table compression policy (whether to enable write compression)	Policy for compressing data online. This parameter takes effect only for MOR tables.	Yes
Hudi Table Clean Policy (Reserved Submissions)	Number of submissions reserved during clearance	1
Hudi Table Archiving Policy (Minimum Retention Submissions)	Minimum number of submissions retained during archiving	1

General Configuration		
Hudi Table Archiving Policy (Maximum Number of Retained Submissions)	Maximum number of submissions retained during archiving	100
Hudi table options	Custom parameters for creating a Hudi table. The parameters take effect in options, for example, **primary key**, **combineKey**, or **index**.	-

5.4.12 To MRS ClickHouse

If the destination link of a job is an **MRS ClickHouse link**, configure the destination job parameters based on **Table 5-40**.

NOTE

If the source link of the job is an MRS ClickHouse, DWS, or Hive link:

- If the int or float fields are null, set the field type to **nullable()** when creating an MRS ClickHouse table. Otherwise, the value written to MRS ClickHouse is **0**.
- Check whether the destination table engine is ReplicatedMergeTree. This engine has a deduplication mechanism, in which the data to be deduplicated cannot be predicted accurately. If this engine is used, ensure that data is unique. Otherwise, non-unique data will be ignored and not written, or ReplicatedMergeTree will be replaced by other types of table engines such as MergeTree.

Table 5-40 Parameter description

Parameter	Description	Example Value
Schema/ Tablespace	Click the icon next to the text box to select a schema or tablespace.	schema

Parameter	Description	Example Value
Table Name	Destination table name. Click the icon next to the text box. The dialog box for selecting the table is displayed. This parameter can be configured as a macro variable of date and time and a path name can contain multiple macro variables. When the macro variable of date and time works with a scheduled job, the incremental data can be synchronized periodically. For details, see **Incremental Synchronization Using the Macro Variables of Date and Time**. **NOTE** If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job* – *Offset*) rather than (*Actual start time of the CDM job* – *Offset*).	table
Clear Data Before Import	Whether to clear the data in the destination table before data import. The options are as follows: • **Do not clear**: The data in the destination table is not cleared before data import. The imported data is just added to the table. • **Clear all data**: All data is cleared from the destination table before data import. • **Clear part of data**: Part of the data in the destination table is cleared before data import. If you select **Clear part of data**, you must configure **WHERE Clause** to specify which part will be deleted.	Clear part of data
Whether On Cluster	This parameter is displayed when **Clear Data Before Import** is set to **Clear part of data** or **Clear all data**. If this parameter is set to **Yes**, all or part of data on all the nodes in the cluster will be cleared.	Yes
WHERE Clause	If **Clear Data Before Import** is set to **Clear part of data**, data in the destination table will be deleted based on the WHERE clause after the configuration is complete and before the import starts.	age > 18 and age <= 60

5.4.13 To MongoDB

If the destination link of a job is a **MongoDB link**, configure the destination job parameters based on **Table 5-41**.

Table 5-41 Parameter description

Parameter	Description	Example Value
Database Name	Database to which data is to be imported	mddb
Collection Name	Collection of data to be imported, which is similar to the table name of a relational database. Click the icon next to the text box to go to the page for selecting the table or directly enter a table name. If the desired table is not displayed, check whether the table exists or whether the login account has the permission to query metadata.	COLLECTION
Behavior	Insert operation to be performed during record migration to the MongoDB • **Insert**: Insert file records into a specified set. • **Insert**: Use a specified filter key as the query condition. If a matching record is found in the set, the record is replaced. (If multiple matching records are found, only the first found record is replaced.) Otherwise, the new record will be added. • **Replace**: Use a specified filter key as the query condition. If a matching record is found in the set, the record is replaced. (If multiple matching records are found, only the first found record is replaced.) Otherwise, the new record will not be added.	Add
Prepare for Data Import	MongoDB query statement that needs to be executed before a task is executed **NOTE** • The value is a JSON string that contains two key-value pairs. The first key-value pair specifies the operation type. The key is **type**, and the value can only be **remove** or **drop**. The second key-value pair is the name of the data condition or set to be configured for the operation type. • The execution of the data import preparation statement does not affect the data to be written.	{"type":"remove","json":"{$or:[{Pid:{$gt:'0',$lt:'2'}},{X:{$gt:'50',$lt:'80'}}]}"}

5.5 Configuring Field Mapping

Scenario

- After the job parameters are configured, you can configure field mapping. You can click on the **Map Field** page to customize new fields or click in the **Operation** column to create a field converter.
- If files are migrated between FTP, SFTP, OBS, and HDFS and the migration source's **File Format** is set to **Binary**, files will be directly transferred, free from field mapping.
- In other scenarios, CDM automatically maps fields of the source table and the destination table. You need to check whether the mapping and time format are correct. For example, check whether the source field type can be converted into the destination field type.
- In the auto table creation scenario, you need to add fields to the destination table in advance, and add the fields to the field mapping..

Constraints

- If **Use SQL Statement** is set to **Yes** in the source job configuration, converters cannot be created.
- On the **Map Field** tab page, if CDM fails to obtain all columns by obtaining sample values (for example, when data is exported from HBase, CloudTable, or MongoDB, there is a high probability that CDM failed to obtain all columns), you can click and select **Add a new field** to add new fields to ensure that the data imported to the migration destination is complete.
- When a relational database, Hive, DLI, or MRS Hudi is used as the migration source, sample values cannot be obtained.
- When SQLServer is the destination, fields of the timestamp type cannot be written. You must change their type (for example, to datetime) so that they can be written.
- Column names are displayed when the source of the migration job is OBS, CSV files are to be migrated, and parameter **Extract first row as columns** is set to **Yes**.
- Field mapping is not involved when the binary format is used to migrate files to files.
- In the automatic table creation scenario, you need to manually add fields to the destination table in advance and then add fields to the field mapping.
- After a field is added, its sample value is not displayed on the console. This does not affect the field value transmission. CDM directly writes the field value to the destination end.
- If the field mapping is incorrect, you can adjust the field mapping by dragging fields or clicking to map fields in batches.
- If the data is imported to DWS, you need to select the distribution columns in the destination fields. You are advised to select the distribution columns according to the following principles:

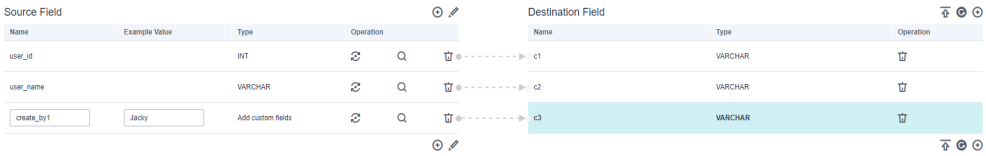
a. Use the primary key as the distribution column.
b. If multiple data segments are combined as primary keys, specify all primary keys as the distribution column.
c. In the scenario where no primary key is available, if no distribution column is selected, DWS uses the first column as the distribution column by default. As a result, data skew risks exist.

- If a source field type is not supported, convert the field type to a type supported by CDM by referring to **Converting Unsupported Data Types**.

Adding a Field

You can click on the **Map Field** page and select **Add** to customize a new field. This field is usually used to mark the database source to ensure the integrity of the data imported to the migration destination.

Figure 5-11 Field mapping



Currently, the following field types are supported:

- **Constant Parameter**

  Constant parameters are fixed parameters and do not need to be reconfigured. For example, **lable** = **friends** is used to identify a constant value.

- **Variables**

  You can use variables such as time macros, table name macros, and version macros to mark database source information. The variable syntax is ${variable}, where **variable** indicates a variable. For example, **input_time** = **${timestamp()}** indicates the timestamp of the current time.

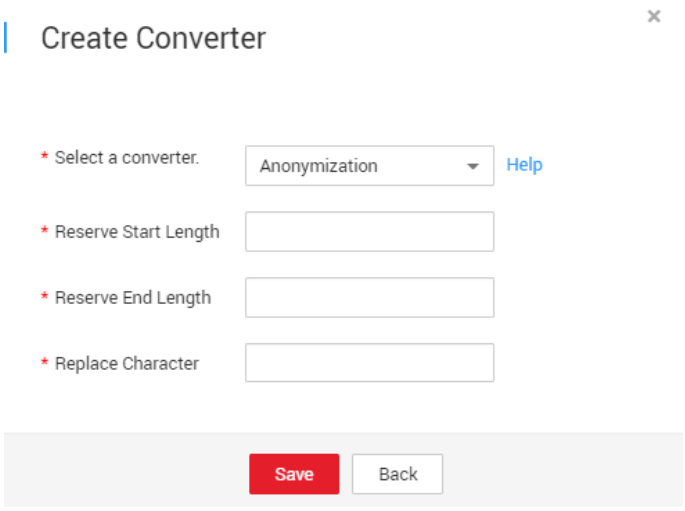
- Expression

  You can use the expression language to dynamically generate parameter values based on the running environment. The expression syntax is #{expr}, where **expr** indicates an expression. For example, **time** = **#{DateUtil.now()}** is used to identify the current date string.

Creating a Converter

CDM supports field conversion. Click and then click **Create Converter**.

Figure 5-12 Creating a converter



CDM can convert fields during migration. Currently, the following field converters are supported:

- **Anonymization**

  This converter is used to hide key information about the character string. For example, if you want to convert **12345678910** to **123****8910**, configure the parameters as follows:

  - Set **Reserve Start Length** to **3**.
  - Set **Reserve End Length** to **4**.
  - Set **Replace Character** to *****.

- **Trim**

  This converter is used to automatically delete the spaces before and after a string. No parameters need to be configured.

- **Reverse string**

  This converter is used to automatically reverse a string. For example, reverse **ABC** into **CBA**. No parameters need to be configured.

- **Replace string**

  This converter is used to replace a character string. You need to configure the object to be replaced and the new value.

- **Remove line break**

  This converter is used to delete the newline characters, such as \n, \r, and \r\n from the field.

- **Expression conversion**

  This converter uses the JSP expression language (EL) to convert the current field or a row of data. The JSP EL is used to create arithmetic and logical expressions. Within a JSP EL expression, you can use integers, floating point numbers, strings, the built-in constants **true** and **false** for boolean values, and **null**.

  - The expression supports the following environment variables:

- **value**: indicates the current field value.
- **row**: indicates the current row, which is an array type.

– The expression supports the following Utils:

i. If the field is of the string type, convert all character strings into lowercase letters, for example, convert **aBC** to **abc**.

Expression: StringUtils.lowerCase(value)

ii. Convert all character strings of the current field to uppercase letters.

Expression: StringUtils.upperCase(value)

iii. Convert the format of the first date field from 2018-01-05 15:15:05 to 20180105.

Expression: DateUtils.format(DateUtils.parseDate(row[0],"yyyy-MM-dd HH:mm:ss"),"yyyyMMdd")

iv. Convert a timestamp to a date string in *yyyy-MM-dd hh:mm:ss* format, for example, convert **1701312046588** to **2023-11-30 10:40:46**.

Expression: DateUtils.format(NumberUtils.toLong(value),"yyyy-MM-dd HH:mm:ss")

v. Convert a date string in the yyyy-MM-dd hh:mm:ss format to a timestamp.

Expression: DateUtils.getTime(DateUtils.parseDate(value,"yyyy-MM-dd hh:mm:ss"))

vi. If the field value is a date string in *yyyy-MM-dd* format, extract the year from the field value, for example, extract **2017** from **2017-12-01**.

Expression: StringUtils.substringBefore(value,"-")

vii. If the field value is of the numeric type, convert the value to a new value which is two times greater than the original value:

Expression: value*2

viii. Convert the field value **true** to **Y** and other field values to **N**.

Expression: value=="true"?"Y":"N"

ix. If the field value is of the string type and is left empty, convert it to **Default**. Otherwise, the field value will not be converted.

Expression: empty value? "Default":value

x. Convert date format **2018/01/05 15:15:05** to **2018-01-05 15:15:05**:

Expression: DateUtils.format(DateUtils.parseDate(value,"yyyy/MM/dd HH:mm:ss"),"yyyy-MM-dd HH:mm:ss")

xi. Obtain a 36-bit universally unique identifier (UUID):

Expression: CommonUtils.randomUUID()

xii. If the field is of the string type, capitalize the first letter, for example, convert **cat** to **Cat**.

Expression: StringUtils.capitalize(value)

xiii. If the field is of the string type, convert the first letter to a lowercase letter, for example, convert **Cat** to **cat**.

Expression: StringUtils.uncapitalize(value)

xiv. If the field is of the string type, use a space to fill in the character string to the specified length and center the character string. If the length of the character string is not shorter than the specified length, do not convert the character string. For example, convert **ab** to meet the specified length 4.

Expression: StringUtils.center(value,*4*)

xv. Delete a newline (including **\n**, **\r**, and **\r\n**) at the end of a character string. For example, convert **abc\r\n\r\n** to **abc\r\n**.

Expression: StringUtils.chomp(value)

xvi. If the string contains the specified string, **true** is returned; otherwise, **false** is returned. For example, **abc** contains **a** so that **true** is returned.

Expression: StringUtils.contains(value,"*a*")

xvii. If the string contains any character of the specified string, **true** is returned; otherwise, **false** is returned. For example, **zzabyycdxx** contains either **z** or **a** so that **true** is returned.

Expression: StringUtils.containsAny(value,"*za*")

xviii. If the string does not contain any one of the specified characters, **true** is returned. If any specified character is contained, **false** is returned. For example, **abz** contains one character of **xyz** so that **false** is returned.

Expression: StringUtils.containsNone(value,"*xyz*")

xix. If the string contains only the specified characters, **true** is returned. If any other character is contained, **false** is returned. For example, **abab** contains only characters among **abc** so that **true** is returned.

Expression: StringUtils.containsOnly(value,"*abc*")

xx. If the character string is empty or null, convert it to the specified character string. Otherwise, do not convert the character string. For example, convert the empty character string to null.

Expression: StringUtils.defaultIfEmpty(value,*null*)

xxi. If the string ends with the specified suffix (case sensitive), **true** is returned; otherwise, **false** is returned. For example, if the suffix of **abcdef** is not null, **false** is returned.

Expression: StringUtils.endsWith(value,*null*)

xxii. If the string is the same as the specified string (case sensitive), **true** is returned; otherwise, **false** is returned. For example, after strings **abc** and **ABC** are compared, **false** is returned.

Expression: StringUtils.equals(value,"*ABC*")

xxiii. Obtain the first index of the specified character string in a character string. If no index is found, **-1** is returned. For example, the first index of **ab** in **aabaabaa** is 1.

Expression: StringUtils.indexOf(value,"*ab*")

xxiv. Obtain the last index of the specified character string in a character string. If no index is found, **-1** is returned. For example, the last index of **k** in **aFkyk** is 4.

Expression: StringUtils.lastIndexOf(value,"*k*")

xxv. Obtain the first index of the specified character string from the position specified in the character string. If no index is found, **-1** is returned. For example, the first index of **b** obtained after the index 3 of **aabaabaa** is 5.

Expression: StringUtils.indexOf(value,"*b*",*3*)

xxvi. Obtain the first index of any specified character in a character string. If no index is found, **-1** is returned. For example, the first index of **z** or **a** in **zzabyycdxx.** is 0.

Expression: StringUtils.indexOfAny(value,"*za*")

xxvii. If the string contains any Unicode character, **true** is returned; otherwise, **false** is returned. For example, **ab2c** contains only non-Unicode characters so that **false** is returned.

Expression: StringUtils.isAlpha(value)

xxviii. If the string contains only Unicode characters and digits, **true** is returned; otherwise, **false** is returned. For example, **ab2c** contains only Unicode characters and digits, so that **true** is returned.

Expression: StringUtils.isAlphanumeric(value)

xxix. If the string contains only Unicode characters, digits, and spaces, **true** is returned; otherwise, **false** is returned. For example, **ab2c** contains only Unicode characters and digits, so that **true** is returned.

Expression: StringUtils.isAlphanumericSpace(value)

xxx. If the string contains only Unicode characters and spaces, **true** is returned; otherwise, **false** is returned. For example, **ab2c** contains Unicode characters and digits so that **false** is returned.

Expression: StringUtils.isAlphaSpace(value)

xxxi. If the string contains only printable ASCII characters, **true** is returned; otherwise, **false** is returned. For example, for **!ab-c~**, **true** is returned.

Expression: StringUtils.isAsciiPrintable(value)

xxxii. If the string is empty or null, **true** is returned; otherwise, **false** is returned.

Expression: StringUtils.isEmpty(value)

xxxiii. If the string contains only Unicode digits, **true** is returned; otherwise, **false** is returned.

Expression: StringUtils.isNumeric(value)

xxxiv. Obtain the leftmost characters of the specified length. For example, obtain the leftmost two characters **ab** from **abc**.

Expression: StringUtils.left(value,*2*)

xxxv. Obtain the rightmost characters of the specified length. For example, obtain the rightmost two characters **bc** from **abc**.

Expression: StringUtils.right(value,*2*)

xxxvi. Concatenate the specified character string to the left of the current character string and specify the length of the concatenated character string. If the length of the current character string is not shorter than the specified length, the character string will not be converted. For example, if **yz** is concatenated to the left of **bat** and the length must be 8 after concatenation, the character string is **yzyzybat** after conversion.

Expression: StringUtils.leftPad(value,*8*,"*yz*")

xxxvi. Concatenate the specified character string to the right of the current character string and specify the length of the concatenated character string. If the length of the current character string is not shorter than the specified length, the character string will not be converted. For example, if **yz** is concatenated to the right of **bat** and the length must be 8 after concatenation, the character string is **batyzyzy** after conversion.

Expression: StringUtils.rightPad(value,*8*,"*yz*")

xxxvii. If the field is of the string type, obtain the length of the current character string. If the character string is null, **0** is returned.

Expression: StringUtils.length(value)

xxxviii. If the field is of the string type, delete all the specified character strings from it. For example, delete **ue** from **queued** to obtain **qd**.

Expression: StringUtils.remove(value,"*ue*")

xl. If the field is of the string type, remove the substring at the end of the field. If the specified substring is not at the end of the field, no conversion is performed. For example, remove **.com** at the end of **www.domain.com**.

Expression: StringUtils.removeEnd(value,"*.com*")

xli. If the field is of the string type, delete the substring at the beginning of the field. If the specified substring is not at the beginning of the field, no conversion is performed. For example, delete **www.** at the beginning of **www.domain.com**.

Expression: StringUtils.removeStart(value,"*www.*")

xlii. If the field is of the string type, replace all the specified character strings in the field. For example, replace **a** in **aba** with **z** to obtain **zbz**.

Expression: StringUtils.replace(value,"*a*","*z*")

xliii. If the field is of the string type, replace multiple characters in the character string at a time. For example, replace **h** in **hello** with **j** and **o** with **y** to obtain **jelly**.

Expression: StringUtils.replaceChars(value,"*ho*","*jy*")

xliv. If the string starts with the specified prefix (case sensitive), **true** is returned; otherwise, **false** is returned. For example, **abcdef** starts with **abc**, so that **true** is returned.

Expression: StringUtils.startsWith(value,"*abc*")

xlv. If the field is of the string type, delete all the specified characters at the beginning and end of the field. the field. For example, delete all **x**, **y**, **z**, and **b** from **abcyx** to obtain **abc**.

Expression: StringUtils.strip(value,"*xyz*b")

xlvi. If the field is of the string type, delete all the specified characters at the end of the field, for example, delete the **abc** string at the end of the field.

Expression: StringUtils.stripEnd(value,*"abc"*)

xlvii. If the field is of the string type, delete all the specified characters at the beginning of the field, for example, delete all spaces at the beginning of the field.

Expression: StringUtils.stripStart(value,*null*)

xlviii. If the field is of the string type, obtain the substring after the specified position (the index starts from 0, including the character at the specified position) of the character string. If the specified position is a negative number, calculate the position in the descending order. The first digit at the end is -1. For example, obtain the second character (c) of **abcde** and the string after it, that is, **cde**.

Expression: StringUtils.substring(value,*2*)

xlix. If the field is of the string type, obtain the substring in a specified range (the index starts from 0, including the character at the start and excluding the character at the end). If the range is a negative number, calculate the position in the descending order. The first digit at the end is -1. For example, obtain the string between the second character (c) and fourth character (e) of **abcde**, that is, **cd**.

Expression: StringUtils.substring(value,*2*,4)

l. If the field is of the string type, obtain the substring after the first specified character. For example, obtain the substring after the first **b** in **abcba**, that is, **cba**.

Expression: StringUtils.substringAfter(value,"*b*")

li. If the field is of the string type, obtain the substring after the last specified character. For example, obtain the substring after the last **b** in **abcba**, that is, **a**.

Expression: StringUtils.substringAfterLast(value,"*b*")

lii. If the field is of the string type, obtain the substring before the first specified character. For example, obtain the substring before the first **b** in **abcba**, that is, **a**.

Expression: StringUtils.substringBefore(value,"*b*")

liii. If the field is of the string type, obtain the substring before the last specified character. For example, obtain the substring before the last **b** in **abcba**, that is, **abc**.

Expression: StringUtils.substringBeforeLast(value,"*b*")

liv. If the field is of the string type, obtain the substring nested within the specified string. If no substring is found, **null** is returned. For example, obtain the substring between **tag** in **tagabctag**, that is, **abc**.

Expression: StringUtils.substringBetween(value,"*tag*")

lv. If the field is of the string type, delete the control characters (char≤32) at both ends of the character string, for example, delete the spaces at both ends of the character string.

Expression: StringUtils.trim(value)

lvi. Convert the character string to a value of the byte type. If the conversion fails, **0** is returned.

Expression: NumberUtils.toByte(value)

lvii. Convert the character string to a value of the byte type. If the conversion fails, the specified value, for example, **1**, is returned.

Expression: NumberUtils.toByte(value,*1*)

lviii. Convert the character string to a value of the double type. If the conversion fails, **0.0d** is returned.

Expression: NumberUtils.toDouble(value)

lix. Convert the character string to a value of the double type. If the conversion fails, the specified value, for example, **1.1d**, is returned.

Expression: NumberUtils.toDouble(value,*1.1d*)

lx. Convert the character string to a value of the float type. If the conversion fails, **0.0f** is returned.

Expression: NumberUtils.toFloat(value)

lxi. Convert the character string to a value of the float type. If the conversion fails, the specified value, for example, **1.1f**, is returned.

Expression: NumberUtils.toFloat(value,*1.1f*)

lxii. Convert the character string to a value of the int type. If the conversion fails, **0** is returned.

Expression: NumberUtils.toInt(value)

lxiii. Convert the character string to a value of the int type. If the conversion fails, the specified value, for example, **1**, is returned.

Expression: NumberUtils.toInt(value,*1*)

lxiv. Convert the character string to a value of the long type. If the conversion fails, **0** is returned.

Expression: NumberUtils.toLong(value)

lxv. Convert the character string to a value of the long type. If the conversion fails, the specified value, for example, **1L**, is returned.

Expression: NumberUtils.toLong(value,*1L*)

lxvi. Convert the character string to a value of the short type. If the conversion fails, **0** is returned.

Expression: NumberUtils.toShort(value)

lxvii. Convert the character string to a value of the short type. If the conversion fails, the specified value, for example, **1**, is returned.

Expression: NumberUtils.toShort(value,*1*)

lxviii. Convert the IP string to a value of the long type, for example, convert **10.78.124.0** to **172915712**.

Expression: CommonUtils.ipToLong(value)

lxix. Read an IP address and physical address mapping file from the network, and download the mapping file to the map collection. *url* indicates the address for storing the IP mapping file, for example, **http://10.114.205.45:21203/sqoop/IpList.csv**.

Expression: HttpsUtils.downloadMap("*url*")

lxx. Cache the IP address and physical address mappings and specify a key for retrieval, for example, **ipList**.

Expression: CommonUtils.setCache("*ipList*",HttpsUtils.downloadMap("*url*"))

lxxi. Obtain the cached IP address and physical address mappings.

Expression: CommonUtils.getCache("*ipList*")

lxxii. Check whether the IP address and physical address mappings are cached.

Expression: CommonUtils.cacheExists("*ipList*")

lxxiii. Based on the specified offset type (month/day/hour/minute/second) and offset (positive number indicates increase and negative number indicates decrease), convert the time in the specified format to a new time, for example, add 8 hours to **2019-05-21 12:00:00**.

Expression: DateUtils.getCurrentTimeByZone("*yyyy-MM-dd HH:mm:ss*",value, "*hour*", *8*)

lxxiv. If the value is empty or null, "aaa" is returned. Otherwise, **value** is returned.

Expression: StringUtils.defaultIfEmpty(value,*"aaa"*)

Special Links

- If the source link is a DLI link, and the destination link is a DWS link, fields of the tinyint type of the DLI link are mapped to fields of the smallint type of the DWS link.
- If the source link is a Hudi link, and the destination link is a DWS link, fields of the Double type of the Hudi link are mapped to fields of the Float type of the DWS link.

5.6 Scheduling Job Execution

CDM supports scheduled execution of table/file migration jobs by minute, hour, day, week, and month. This section describes how to configure scheduled job parameters.

NOTE

- When configuring scheduled jobs, do not set the same scheduled time for different jobs. Instead, set different times to avoid exceptions.
- If you use DataArts Studio DataArts Factory to schedule the CDM migration job and configure this parameter, both configurations take effect. To ensure unified service logic and avoid scheduling conflicts, enable job scheduling in DataArts Factory and do not configure a scheduled task for the job in DataArts Migration.
- The scheduled execution function uses the Java Quartz timer, which is similar to the Cron expression configuration. It parses the minute, hour, day, and month of the start time, and constructs a cronb expression.

  For example, in the daily scheduling mode where the interval is set to 1 day: if the current time is 2022-10-14 12:00 and the start time is set to 2022-10-14 00:00, the job is executed at 2022-10-15 00:00; if the current time is 2022-10-14 12:00 and the start time is set to 2022-10-14 00:00, the job is executed at 2022-10-15 00:00.

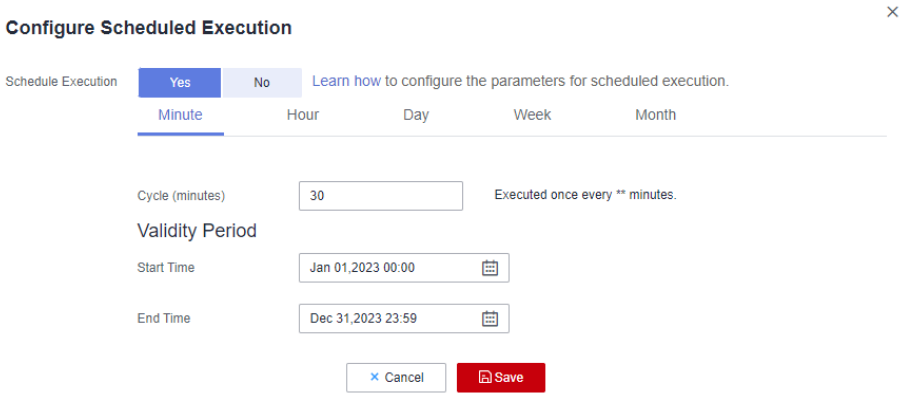
  In the daily scheduling mode where the interval is set to 2 days: if the current time is 2022-10-14 12:00 and the start time is set to 2022-10-14 00:00, the job is executed at 2022-10-16 00:00; if the current time is 2022-10-14 12:00 and the start time is set to 2022-10-14 00:00, the job is executed at 2022-10-16 00:00.

Scheduling Job Execution by Minute

CDM allows jobs to be executed every several minutes. It is recommended that the cycle be at least 5 minutes.

- **Start Time**: indicates the time when the scheduled configuration takes effect, or the first time when the job is automatically executed.
- **Cycle (minutes)**: indicates the interval when a job is executed starting from the start time.
- **End Time**: This parameter is optional. If it is not set, the scheduled job keeps being automatically executed. If it is set, the scheduled job will be automatically stopped at the end time.

Figure 5-13 Scheduling job execution by minute



For example, the settings shown in **Figure 5-13** mean that the job will be automatically executed at 00:00 on January 1, 2023 for the first time at a cycle of 30 minutes until 23:59 on December 31, 2023.

Scheduling Job Execution by Hour

CDM allows jobs to be executed every several hours.

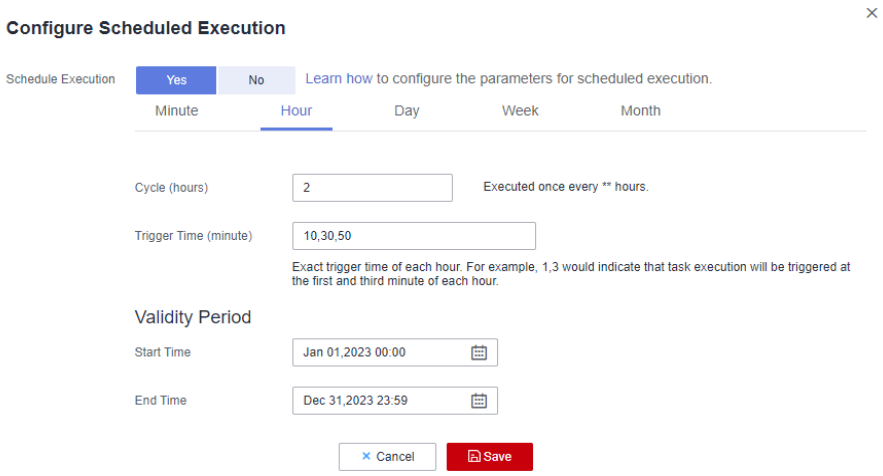
- **Cycle (hours)**: indicates the interval when a job is automatically executed.
- **Trigger Time (minute)**: indicates the exact time in each hour when a scheduled task is triggered. The value ranges from 0 to 59. You can set a maximum of 60 values and use commas (,) to separate these values. However, the values must be unique.

  If the trigger time is not within the validity period, the system selects a trigger time closest to the validity period for the scheduled job to be automatically executed at the first time. The following gives an example:

  - **Start Time**: **1:20**
  - **Cycle (hours)**: **3**
  - **Trigger Time (minute)**: **10**
- **Validity Period**: includes **Start Time** and **End Time**.
  - **Start Time**: indicates the time when the scheduled configuration takes effect.

  - **End Time**: This parameter is optional, which indicates the time when the scheduled job is automatically stopped. If this parameter is not set, the scheduled job keeps being automatically executed.

Figure 5-14 Scheduling job execution by hour



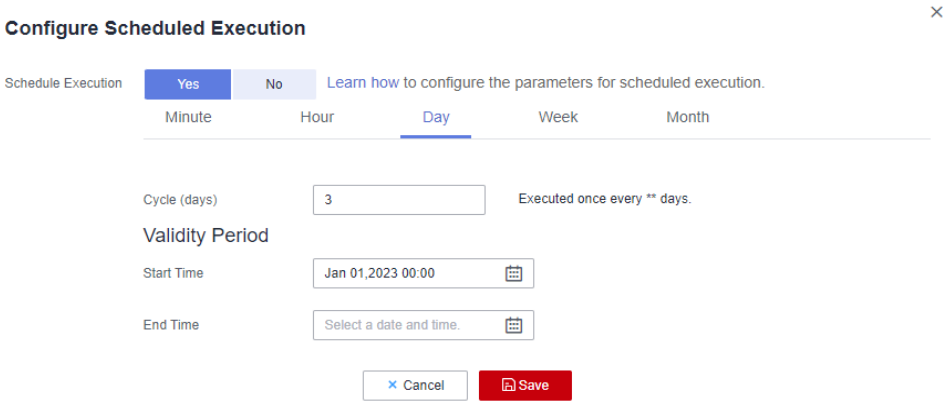
For example, the settings shown in **Figure 5-14** mean that the job will be automatically executed at 00:10 on January 1, 2023 for the first time, at 00:30 for the second time, and at 00:50 for the third time. It will be executed three times every two hours until 23:59 on December 31, 2023.

Scheduling Job Execution by Day

CDM allows jobs to be executed every several days.

- **Cycle (days)**: indicates the interval when a job is executed starting from the start time.
- **Validity Period**: includes **Start Time** and **End Time**.
  - **Start Time**: indicates the time when the scheduled configuration takes effect, or the first time when the job is automatically executed.
  - **End Time**: This parameter is optional, which indicates the time when the scheduled job is automatically stopped. If this parameter is not set, the scheduled job keeps being automatically executed.

Figure 5-15 Scheduling job execution by day



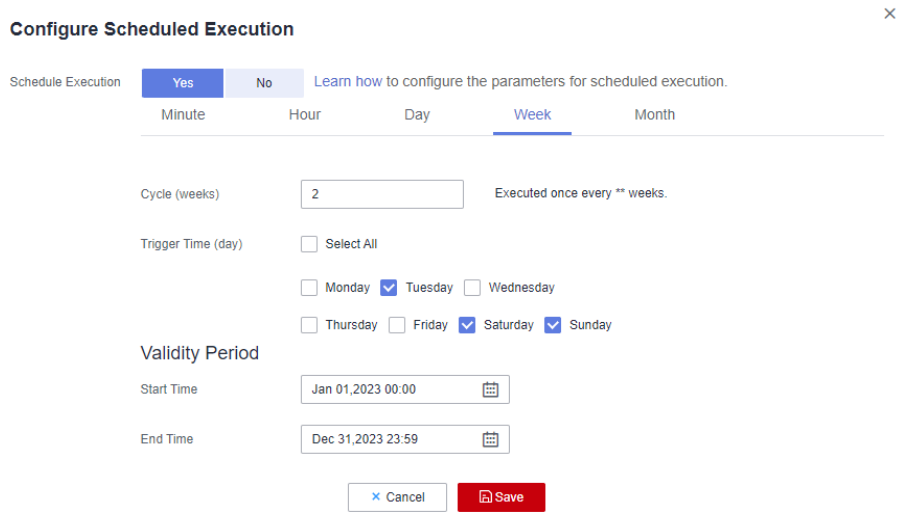
For example, the settings shown in **Figure 5-15** mean that the job will be automatically executed at 00:00 on January 1, 2023 for the first time, and will be executed once every three days. The configuration is valid permanently.

Scheduling Job Execution by Week

CDM allows jobs to be executed every several weeks.

- **Cycle (weeks)**: indicates the interval when a scheduled job is executed starting from the start time.
- **Trigger Time (day)**: You can specify the day of each week when the job is automatically executed. One or more days can be selected at a time.
- **Validity Period**: includes **Start Time** and **End Time**.
  - **Start Time**: indicates the time when the scheduled configuration takes effect.
  - **End Time**: This parameter is optional, which indicates the time when the scheduled job is automatically stopped. If this parameter is not set, the scheduled job keeps being automatically executed.

Figure 5-16 Scheduling job execution by week



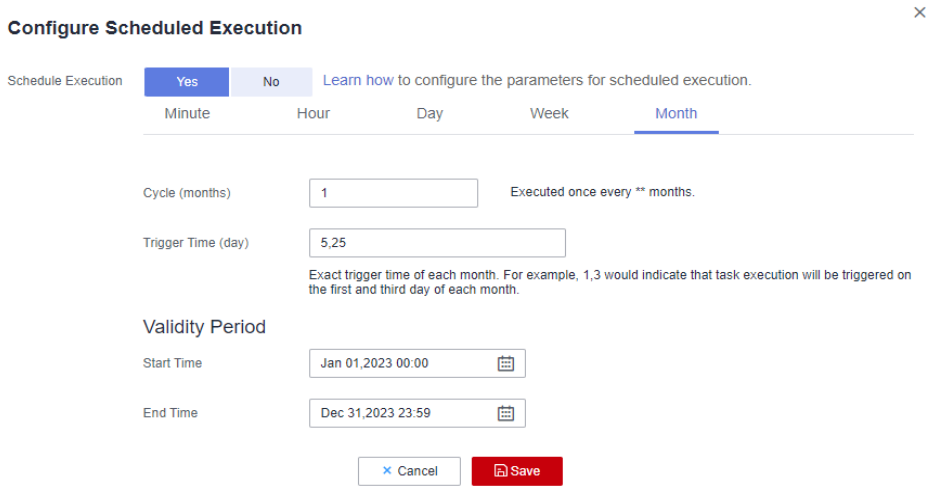
For example, the settings shown in **Figure 5-16** mean that the job will be automatically executed at 00:00 every Tuesday, Saturday, and Sunday every two weeks starting from 00:00 on January 1, 2023 until 23:59 on December 31, 2023.

Scheduling Job Execution by Month

CDM allows jobs to be executed every several months.

- **Cycle (months)**: indicates the interval when a scheduled job is executed starting from the start time.
- **Trigger Time (day)**: indicates the day of each month when the job is executed. The value ranges from 1 to 31. You can set multiple values and use commas (,) to separate these values. However, the values must be unique.
- **Validity Period**: includes **Start Time** and **End Time**.
  - **Start Time**: indicates the time when the scheduled configuration takes effect. The automatic execution time is accurate to hour, minute, and second.
  - **End Time**: This parameter is optional, which indicates the time when the scheduled job is automatically stopped. If this parameter is not set, the scheduled job keeps being automatically executed.

Figure 5-17 Scheduling job execution by month



For example, the settings shown in **Figure 5-17** mean that the job will be automatically executed at 00:00 on the 5th and 25th days of each month starting from 00:00 on January 1, 2023 until 23:59 on December 31, 2023.

5.7 Job Configuration Management

On the **Settings** tab page, you can perform the following operations:

- **Maximum Concurrent Extractors**
- **Scheduled Backup/Restoration**

- **Environment Variables of Job Parameters**

Maximum Concurrent Extractors

Maximum number of concurrent extraction tasks in a cluster

CDM migrates data through data migration jobs. It works in the following way:

1. When data migration jobs are submitted, CDM splits each job into multiple tasks based on the **Concurrent Extractors** parameter in the job configuration.

   NOTE

   Jobs for different data sources may be split based on different dimensions. Some jobs may not be split based on the **Concurrent Extractors** parameter.

2. CDM submits the tasks to the running pool in sequence. Tasks (defined by **Maximum Concurrent Extractors**) run concurrently. Excess tasks are queued.

By setting an appropriate number of concurrent extractors for a job and the maximum number of concurrent extractors for the cluster, you can accelerate migration. You can configure the number of concurrent extractors as follows:

1. You are advised to set **Maximum Concurrent Extractors** to twice the number of vCPUs. For details, see **Table 5-42**.

   Table 5-42 Recommended maximum number of concurrent extractors for a CDM cluster

Flavor	vCPUs/Memory	Recommended Maximum Concurrent Extractors
cdm.large	8 vCPUs, 16 GB	16
cdm.xlarge	16 vCPUs, 32 GB	32
cdm.4xlarge	64 vCPUs, 128 GB	128

2. Configure the number of concurrent extractors based on the following rules:
   a. When data is to be migrated to files, CDM does not support multiple concurrent tasks. In this case, set a single process to extract data.
   b. If each row of the table contains less than or equal to 1 MB data, data can be extracted concurrently. If each row contains more than 1 MB data, it is recommended that data be extracted in a single thread.
   c. Set **Concurrent Extractors** for a job based on **Maximum Concurrent Extractors** for the cluster. It is recommended that **Concurrent Extractors** is less than **Maximum Concurrent Extractors**.
   d. If the destination is DLI, you are advised to set the number of concurrent extractors to 1. Otherwise, data may fail to be written.

Scheduled Backup/Restoration

This function depends on the OBS service.

- Prerequisites

  An OBS link has been created. For details, see **Link to OBS**.

- Scheduled backup

  On the **Job Management** page, click **Settings** and configure **Scheduled Backup** and its related parameters.

  Table 5-43 Scheduled backup parameters

Parameter	Description	Example Value
Scheduled Backup	Whether to enable automatic backup. This function is used to back up jobs but not links.	Enable
Backup Policy	• **All jobs**: CDM backs up all table/file migration jobs and entire DB migration jobs regardless of the job statuses. However, historical jobs are not backed up. • **All jobs by groups**: You select one or more job groups to back up.	All jobs
Backup Cycle	Select the backup cycle. • **Day**: The backup is performed daily at 00:00:00. • **Week**: The backup is performed at 00:00:00 every Monday. • **Month**: The backup is performed at 00:00:00 on the first day of each month.	Day
OBS Link for Writing Backups	Link used to back up jobs to OBS buckets. Select a link you have created on the **Links** page.	obslink
OBS Bucket	OBS bucket where backup files are stored	cdm
Backup Data Directory	Directory where backup files are stored	/cdm-bk/

- Restoring jobs

  If automatic backup has been performed, the backup list is displayed on the **Configuration Management** tab page. The OBS buckets where the backup files reside, backup paths, and backup time are displayed.

  You can click **Restore Backup** in the **Operation** column of the backup list to restore the CDM jobs.

Environment Variables of Job Parameters

When creating a migration job on CDM, the parameter (such as the OBS bucket name or file path) that can be manually configured, a field in a parameter, or a

character in a field can be configured as a global variable, so that you can change parameter values in batches, or batch replace certain characters after jobs are exported or imported.

The following describes how to batch replace the OBS bucket name in a migration job.

1. On the **Job Management** page, click the **Configuration Management** tab and configure environment variables.

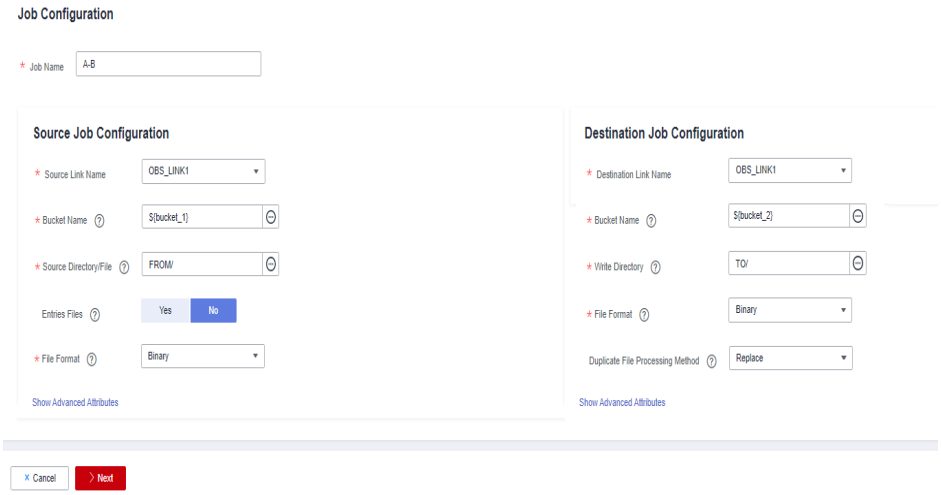
   ```
   bucket_1=A
   bucket_2=B
   ```

   Variable **bucket_1** indicates bucket A, and variable **bucket_2** indicates bucket B.

2. On the page for creating a CDM migration job, migrate data from bucket A to bucket B.

   Set the source bucket name to **${bucket_1}** and destination bucket name to **${bucket_2}**.

   Figure 5-18 Setting the bucket names to environment variables

3. If you want to migrate data from bucket C to bucket D, you do not need to change the job parameters. You only need to change the environment variables on the **Configuration Management** tab page as follows:

   ```
   bucket_1=C
   bucket_2=D
   ```

5.8 Managing a Single Job

Existing CDM jobs can be viewed, modified, deleted, started, and stopped. This section describes how to view and modify a job.

Viewing a Job

- **Viewing job status**

  The job status can be **New**, **Pending**, **Booting**, **Running**, **Failed**, **Succeeded**, or **Stopped**.

  Pending indicates that the job is waiting to be scheduled by the system, and **Booting** indicates that the data to be migrated is being analyzed.

- **Viewing the historical records**

  On the **Historical Record** page, you can view job execution records, read/write statistics, and job execution logs.

- **Viewing job logs**

  On the **Historical Record** page, you can view all logs of a job.

  Alternatively, in the **Operation** column, choose **More** > **Log** to view the latest logs of the job.

- **Viewing the JSON file of a job**

  You can directly edit the JSON file of a job, which is equivalent to modifying the parameter settings of the job.

- **Querying the job statistics**

  You can open the preview window of a configured database job and view up to 1,000 pieces of data. By comparing the number of data records of the migration source and destination, you can check whether the migration was successful and whether data was lost.

Modifying a Job

- **Modifying the job parameters**

  You can reconfigure job parameters, but you cannot reselect source and destination links.

- **Editing the JSON file of a job**

  You can directly edit the JSON file of a job, which is equivalent to modifying the parameter settings of the job.

Procedure

Step 1 Log in to the management console and choose **Service List** > **Cloud Data Migration**. In the left navigation pane, choose **Cluster Management**. Locate the target cluster and click **Job Management**.

Step 2 Click **Historical Jobs** to view all historical jobs executed in the latest month.

Step 3 Click **Table/File Migration**. The job list is displayed. You can perform the following operations on a single job:

- Modify the job parameters: Click **Edit** in the **Operation** column to modify the job parameters.
- Run the job: Click **Run** in the **Operation** column to manually start the job.
- View the historical records: Click **Historical Record** in the **Operation** column. On the **Historical Record** page that is displayed, view the job's historical execution records and read/write statistics. Click **Log** to view the job logs.
- Delete the job: Choose **More** > **Delete** in the **Operation** column to delete the job.
- Stop the job: Choose **More** > **Stop** in the **Operation** column to stop the job.
- View the job JSON: Choose **More** > **View Job JSON** in the **Operation** column to view the job JSON.
- Edit the job JSON: Choose **More** > **Edit Job JSON** in the **Operation** column to edit the job JSON files, which is similar to modify the job parameters.

- Configure a scheduled job: Locate a job and choose **More** > **Configure Scheduled Execution**. You can set the cycle for periodically executing the job. For details, see **Scheduling Job Execution**.
- View logs: Locate a job, click **More** in the **Operation** column, and select **Log** to view the latest log of the job.

  You can also view all logs of the job on the **Historical Record** page.
- Retry the job: Locate a failed job, click **More** in the **Operation** column, and select **Retry**. The job will be automatically retried three times.

Step 4 After the modification, click **Save** or **Save and Run**.

----End

5.9 Managing Jobs in Batches

Scenario

This section describes how to manage CDM table/file migration jobs in batches. The following operations are involved:

- Manage jobs by group.
- Run jobs in batches.
- Delete jobs in batches.
- Export jobs in batches.
- Import jobs in batches.

You can export and import jobs in batches in the following scenarios:

- Job migration between CDM clusters: You can migrate jobs from a cluster of an earlier version to a new version.
- Job backup: You can stop or delete CDM clusters to reduce costs. In this case, you can export the job scripts in batches and save them, and create a cluster and import the job scripts if necessary.
- Batch job creation: You can manually create a job and export the job configuration file in JSON format. Copy the content in the JSON file to the same file or new files, and then import the file/files to CDM to create jobs in batches.

Procedure

Step 1 Log in to the management console and choose **Service List** > **Cloud Data Migration**. In the left navigation pane, choose **Cluster Management**. Locate the target cluster and click **Job Management**.

Step 2 Click **Table/File Migration**. The job list is displayed. You can perform the following batch operations:

- **Manage jobs by group.**

  CDM allows users to add, modify, search for, and delete job groups. When a group is deleted, all jobs in the group are deleted.

  In the third step of creating a job, if jobs have been assigned to different groups, you can display, start, or export jobs by group.

- **Run jobs in batches.**

  After selecting one or more jobs, click **Run** to start these jobs in batches.

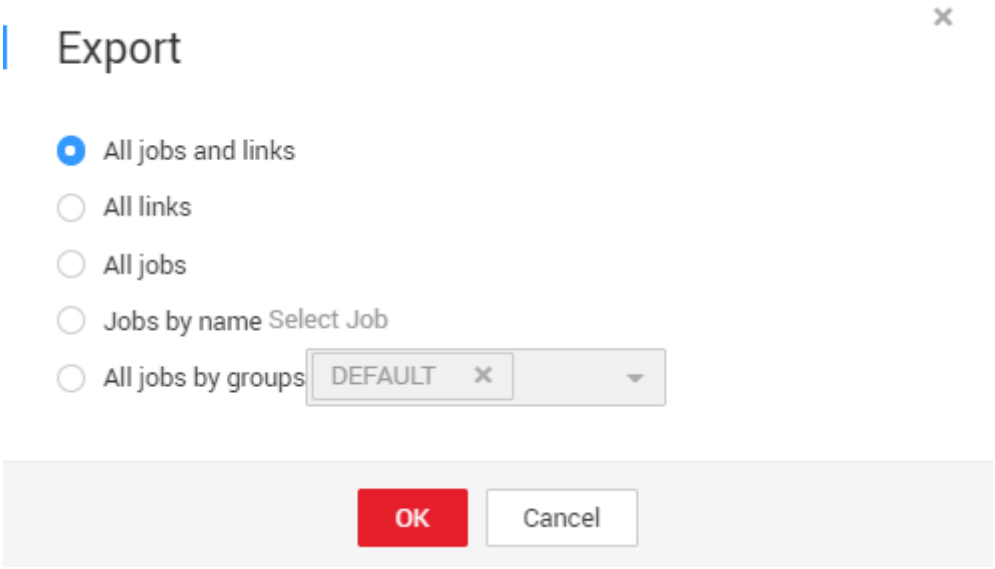
- **Delete jobs in batches.**

  After selecting one or more jobs, click **Delete** to delete these jobs in batches.

- **Export jobs in batches.**

  Click **Export**.

  Figure 5-19 Export

  - **All jobs and links**: Export all jobs and links at a time.
  - **All jobs**: Export all jobs at a time.
  - **All links**: Export all links at a time.
  - **Jobs by name**: Select the jobs to export and click **OK**.
  - **All jobs by groups**: Select the group to export and click **OK**.

  Exported jobs are stored in JSON files, which can be used as backups or imported to other clusters.

  NOTE

  For security purposes, no link password is exported when jobs are exported. All passwords are replaced by *Add password here*.

- **Import jobs in batches.**

  Click **Import** and select the import format (text file or JSON).

  - **By JSON string**: Job files to be imported must be in JSON format and the file size cannot exceed 1 MB. If the job files to be imported are exported from CDM, edit the JSON files before importing them to CDM. Replace *Add password here* with the correct link passwords.
  - **By text file**: This mode can be used when the local JSON files cannot be uploaded properly. Paste the JSON strings for the jobs into the text box.

NOTE

Existing jobs cannot be overwritten during the import.

----End

6 Viewing Traces

6.1 Viewing Traces

Overview

You can use Cloud Trace Service (CTS) to record key operation events related to CDM. The events can be used in various scenarios such as security analysis, compliance audit, resource management, and problem locating.

After you enable CTS, the system starts to record the CDM operations. The management console of CTS stores the traces of the latest seven days.

Prerequisites

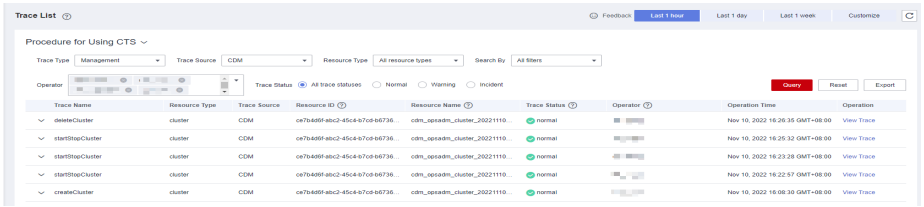
CTS has been enabled. For details about how to enable it, see **Enabling CTS**.

Procedure

1. Log in to the management console and choose **Cloud Trace Service** from the service list.
2. The trace list is displayed by default. You can filter traces.

   You can select **CDM** for **Trace Source** to filter out CDM traces.

   Figure 6-1 CDM traces

3. Click ⌄ on the left of a trace to expand its details.
4. Click **View Trace** in the **Operation** column to view the trace structure details.

   For more information about CTS, see *Cloud Trace Service User Guide*.

6.2 Key CDM Operations Recorded by CTS

CTS provides records of operations on cloud service resources. With CTS, you can query, audit, and backtrack those operations.

Table 6-1 Key operations recorded by CTS

Operation	Resource Type	Trace Name
Creating a cluster	cluster	createCluster
Deleting a cluster	cluster	deleteCluster
Modifying cluster configurations	cluster	modifyCluster
Starting a cluster	cluster	startCluster
Restarting a cluster	cluster	restartCluster
Importing a job	cluster	clusterImportJob
Binding an EIP	cluster	bindEip
Unbinding an EIP	cluster	unbindEip
Creating a link	link	createLink
Modifying a link	link	modifyLink
Testing a link	link	verifyLink
Deleting a link	link	deleteLink
Creating a job	job	createJob
Modifying a job	job	modifyJob
Deleting a job	job	deleteJob
Starting a job	job	startJob
Stopping a job	job	stopJob

7 Key Operation Guide

7.1 Incremental Migration

7.1.1 Incremental File Migration

CDM supports incremental migration of file systems. After full migration is complete, all new files or only specified directories or files can be exported.

Currently, CDM supports the following incremental migration modes:

1. **Exporting the files in a specified directory**
   - Application scenarios: The migration source is a file system (OBS/HDFS/FTP/SFTP). In incremental migration, only the specified files are written to the migration destination. The existing records are not updated or deleted.
   - Key configurations: **File/Path Filter** and Schedule Execution
   - Prerequisites: The source directory or file name contains the time field.
2. **Exporting the files modified after the specified time point**
   - Application scenarios: The migration source is a file system (OBS/HDFS/FTP/SFTP). The specified time point refers to the time when the file is modified. CDM migrates the files modified at or after the specified time point.
   - Key configurations: **Time Filter** and Schedule Execution
   - Prerequisites: None

NOTE

If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).

File/Path Filter

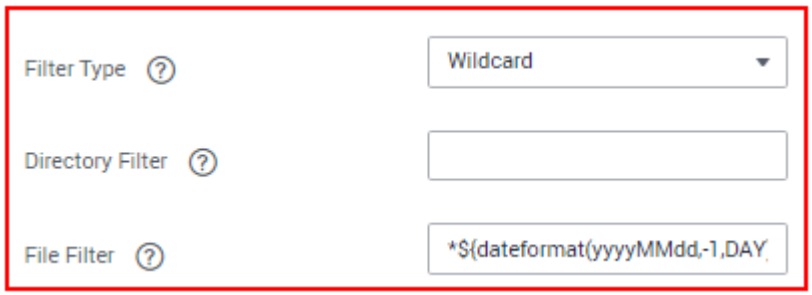
- Parameter position: When creating a table/file migration job, if the migration source is a file system, set **Filter Type** in advanced attributes of **Source Job Configuration** to **Wildcard** or **Regular expression**.
- Parameter principle: If you select **Wildcard** for **Filter Type**, CDM filters files or paths based on the configured wildcard character and migrates only files or paths that meet the specified condition.
- Example configurations:

  Suppose that the source file name contains the date and time field, such as **2017-10-15 20:25:26**, the **/opt/data/file_20171015202526.data** file is generated. Set the parameters as follows:

  a. **Filter Type**: Select **Wildcard**.

  b. **File Filter**: Enter **"*${dateformat(yyyyMMdd,-1,DAY)}*"**, which is the format of the macro variables of date and time supported by CDM. For details, see **Using Macro Variables of Date and Time**.

  Figure 7-1 Filtering files

  c. Schedule Execution: Set **Cycle (days)** to **1**.

In this way, you can import the files generated in the previous day to the destination directory every day to implement incremental synchronization.

In incremental file migration, **Path Filter** is used in the same way as **File Filter**. The path name must contain the time field. In this case, all files in the specified path can be synchronized periodically.

Time Filter

- Parameter position: When creating a table/file migration job, if the migration source is a file system, set select **Yes** for **Time Filter**.
- Parameter principle: After you specify the start time and end time, only files that are modified between the start time (included) and end time (excluded) will be migrated.
- Example configurations:

  For example, if you want CDM to synchronize only the files generated from January 1, 2021 to January 1, 2022 to the destination, configure the following parameters:

  a. **Time Filter**: select **Yes**.

  b. **Minimum Timestamp**: Enter a value in the format of *yyyy-MM-dd HH:mm:ss*, such as **2021-01-01 00:00:00**.

c. **Maximum Timestamp**: Enter a value in the format of *yyyy-MM-dd HH:mm:ss*, such as **2022-01-01 00:00:00**.

Figure 7-2 Time Filter

In this way, the CDM job migrates only the files generated from January 1, 2021 to January 1, 2022, and performs incremental synchronization next time it is started.

7.1.2 Incremental Migration of Relational Databases

CDM supports incremental migration of relational databases. After a full migration is complete, data in a specified period can be incrementally migrated. For example, data added on the previous day can be exported at 00:00:00 every day.

- **Migrating incremental data within a specified period of time**
  - Application scenarios: The source end is a relational database. The destination end can be of any type.
  - Key configurations: **WHERE Clause** and Schedule Execution
  - Prerequisites: The data table contains a date and time field or timestamp field.

In incremental migration, only the specified data is written to the data table. The existing records are not updated or deleted.

NOTE

If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).

WHERE Clause

- Parameter position: When creating a table/file migration job, if the source end is a relational database, the **Where Clause** parameter is available in the advanced attributes of **Source Job Configuration**.
- Parameter principle: Set **WHERE Clause** to an SQL statement, for example, **age > 18 and age <= 60**, CDM exports only the data that meets the SQL statement requirement. If **WHERE Clause** is not specified, the entire table is exported.

  Where Clause can be set to **macro variables of date and time**. When the data table contains the **date** or **timestamp** field, **Where Clause** and Schedule Execution can be used together to extract data of a specified date.

- Example configurations:

  Suppose that the database table contains column **DS** indicating the time, the value type of the column is **varchar(30)**, and the inserted time format is similar to *2017-xx-xx*. See **Figure 7-3**. Set the parameters as follows:

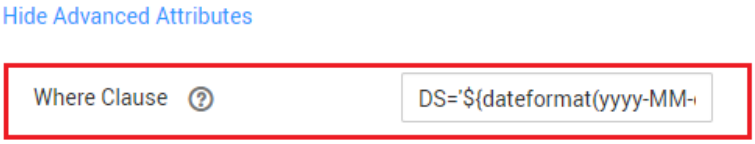
  Figure 7-3 Table data

  SELECT * FROM SQOOP.CDM_20171016

	FOO	BAR	DS
1	5	snap	2017-05-01
2	5	snap	2017-05-01
3	1	google	2017-05-02
4	4	oracle	2017-05-02
5	6	amd	2017-05-02
6	7	nvda	2017-05-02
7	1	google	2017-05-02
8	4	oracle	2017-05-02
9	6	amd	2017-05-02
10	7	nvda	2017-05-02
11	2	facebook	2017-10-15
12	3	tesla	2017-10-15
13	2	facebook	2017-10-15
14	3	tesla	2017-10-15

  a. **WHERE Clause**: Set this parameter to **DS='${dateformat(yyyy-MM-dd,-1,DAY)}'**.

  Figure 7-4 WHERE Clause

  b. Scheduling job execution: Set **Cycle (days)** to **1** and **Start Time** to **00:00:00**.

  In this way, all data generated on the previous day can be exported at 00:00:00 every day. **WHERE Clause** can be configured to various **macro variables of date and time**. You can use the macro variables of date and time and scheduled jobs with specified cycle of minutes, hours, days, weeks, or months together to automatically export data at a specific time.

7.1.3 HBase/CloudTable Incremental Migration

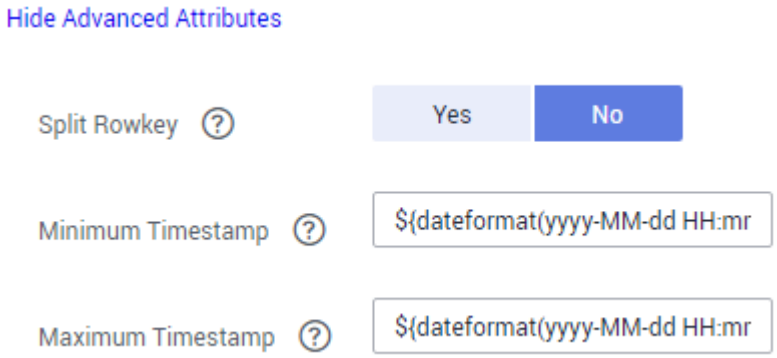
You can use CDM to export data in a specified period of time from HBase (including MRS HBase, FusionInsight HBase, and Apache HBase) and CloudTable. The CDM scheduled jobs can be used together to implement incremental migration of HBase and CloudTable.

NOTE

If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).

When creating a table/file migration job and selecting the link to HBase or CloudTable as the source link, you can set the time range in advanced attributes.

Figure 7-5 Time range



- Start time (including the value) for extracting data. The format is *yyyy-MM-dd HH:mm:ss*. Only the data generated at the specified time and later is extracted.
- End time (excluding the value) for extracting data. The format is *yyyy-MM-dd HH:mm:ss*. Only the data generated before the time point is extracted.

The two parameters can be set to **macro variables of date and time**. Examples are as follows:

- If **Minimum Timestamp** is set to **${dateformat(yyyy-MM-dd HH:mm:ss, -1, DAY)}**, only the data generated after the day before is exported.
- If **Maximum Timestamp** is set to **${dateformat(yyyy-MM-dd HH:mm:ss)}**, only the data generated before the specified time point is exported.

If both parameters are configured, CDM exports only the data generated on the previous day. In addition, if the job is configured to execute at 00:00:00 every day, the data generated every day can be incrementally synchronized.

7.1.4 MongoDB/DDS Incremental Migration

By using CDM, you can export MongoDB or DDS data within a specified period. With the scheduled jobs of CDM, you can implement incremental migration of MongoDB and DDS.

NOTE

If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).

When creating a table/file migration job and selecting the link to MongoDB or DDS as the source link, you can set the query filters in advanced attributes.

Figure 7-6 Setting query filters

You can set this parameter to a **macro variable of date and time**, for example, **{"ts":{$gte:ISODate("${dateformat(yyyy-MM-dd'T'HH:mm:ss.SSS'Z',-1,DAY)}")}}**, which indicates searching for the values in the **ts** field that are greater than those after time macro conversion, that is, only the data generated after the previous day is exported.

After this parameter is set, CDM exports only the data generated on the previous day. In addition, you can set the job to be executed at 00:00:00 every day, so that the data generated every day can be incrementally synchronized.

7.2 Using Macro Variables of Date and Time

During the creation of table/file migration jobs, CDM supports the macro variables of date and time in the following parameters of the source and destination links:

- Source directory or file
- Source table name
- Directory filter and file filter of the **wildcard** type
- Start time and end time of the **time filter** type
- Partition filter criteria and where clause
- Write directory
- Destination table name

You can use the **${}** macro variable definition identifier to define the macros of the time type. currently, dateformat and timestamp are supported.

By using the macro variables of date and time and scheduled job, you can implement incremental synchronization of databases and files.

NOTE

If you have configured a macro variable of date and time and schedule a CDM job through DataArts Studio DataArts Factory, the system replaces the macro variable of date and time with (*Planned start time of the data development job – Offset*) rather than (*Actual start time of the CDM job – Offset*).

dateformat

dateformat supports two types of parameters:

- **dateformat(format)**

  format indicates the date and time format. For details about the format definition, see the definition in **java.text.SimpleDateFormat.java**.

  For example, if the current date is **2017-10-16 09:00:00**, **yyyy-MM-dd HH:mm:ss** indicates **2017-10-16 09:00:00**.

- dateformat(format, dateOffset, dateType)
  - **format** indicates the format of the returned date.
  - **dateOffset** indicates the date offset.
  - **dateType** indicates the type of the date offset.

    Currently, **dateType** supports SECOND, MINUTE, HOUR, MONTH, YEAR, and DAY.

    NOTE

    Pay attention to the following special scenarios of **MONTH** and **YEAR**:

    - If the date does not exist after the offset, the latest date of the month in the calendar is used.
    - These two offset types cannot be used for the start time and end time in the **Time Filter** parameter of the source and destination jobs.

  For example, if the current date is **2023-03-01 09:00:00**, then:

  - **dateformat(yyyy-MM-dd HH:mm:ss, -1, YEAR)** indicates the year before the current time, that is, **2022-03-01 09:00:00**.
  - **dateformat(yyyy-MM-dd HH:mm:ss, -3, MONTH)** indicates three months before the current time, that is, **2022-12-01 09:00:00**.
  - **dateformat(yyyy-MM-dd HH:mm:ss, -1, DAY)** indicates the day before the current time, that is, **2023-02-28 09:00:00**.
  - **dateformat(yyyy-MM-dd HH:mm:ss, -1, HOUR)** indicates one hour before the current time, that is, **2023-03-01 08:00:00**.
  - **dateformat(yyyy-MM-dd HH:mm:ss, -1, MINUTE)** indicates one minute before the current time, that is, **2023-03-01 08:59:00**.
  - **dateformat(yyyy-MM-dd HH:mm:ss, -1, SECOND)** indicates one second before the current time, that is, **2023-03-01 08:59:59**.

timestamp

timestamp supports two types of parameters:

- **timestamp()**

  Indicates the returned timestamp of the current time, that is, the number of milliseconds that have elapsed since 00:00:00 on January 1, 1970 (1970-01-01 00:00:00 GMT). For example, 1508078516286.

- **timestamp(dateOffset, dateType)**

  Indicates the timestamp returned after time offset. **dateOffset** and **dateType** indicate the date offset and the offset type, respectively.

  For example, if the current date is **2017-10-16 09:00:00**, **timestamp(-10, MINUTE)** indicates that the timestamp generated 10 minutes before the current time point is returned, that is, **1508115000000**.

Macro Variable Definition of Time and Date

Suppose that the current time is **2017-10-16 09:00:00**, then **Table 7-1** describes the macro variable definitions of time and date.

Table 7-1 Macro variable definition of time and date

Macro Variable	Description	Display Effect
${dateformat(yyyy-MM-dd)}	Returns the current date in **yyyy-MM-dd** format.	2017-10-16
${dateformat(yyyy/MM/dd)}	Returns the current date in **yyyy/MM/dd** format.	2017/10/16
${dateformat(yyyy_MM_dd HH:mm:ss)}	Returns the current time in **yyyy_MM_dd HH:mm:ss** format.	2017_10_16 09:00:00
${dateformat(yyyy-MM-dd HH:mm:ss, -1, DAY)}	Returns the current time in **yyyy-MM-dd HH:mm:ss** format. The date is one day before the current day.	2017-10-15 09:00:00
${timestamp()}	Returns the timestamp of the current time, that is, the number of milliseconds that have elapsed since 00:00:00 on January 1, 1970.	1508115600000
${timestamp(-10, MINUTE)}	Returns the timestamp generated 10 minutes before the current time point.	1508115000000
${timestamp(dateformat(yyyyMMdd))}	Returns the timestamp of 00:00:00 of the current day.	1508083200000
${timestamp(dateformat(yyyyMMdd,-1,DAY))}	Returns the timestamp of 00:00:00 of the previous day.	1507996800000
${timestamp(dateformat(yyyyMMddHH))}	Returns the timestamp of the current hour.	1508115600000

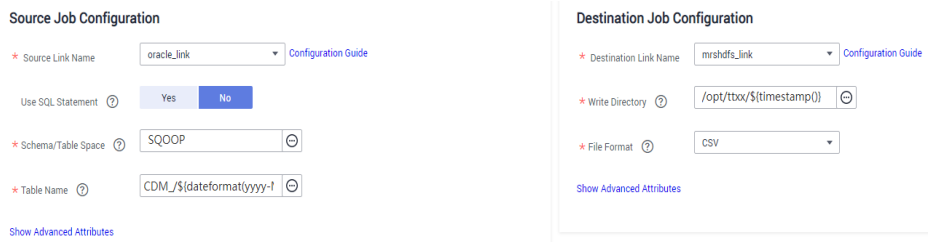
Time and Date Macro Variables of Paths and Table Names

Figure 7-7 shows an example. If:

- **Table Name** under **Source Link Configuration** is set to **CDM_/${dateformat(yyyy-MM-dd)}**.
- **Write Directory** under **Destination Link Configuration** is set to **/opt/ttxx/${timestamp()}**.

After the macro definition conversion, this job indicates that data in table **SQOOP.CDM_20171016** in the Oracle database is migrated to the **/opt/ttxx/1508115701746** directory of the HDFS server.

Figure 7-7 Setting **Table Name** and **Write Directory** to a time and date macro variable



Currently, a table name or path name can contain multiple macro variables. For example, **/opt/ttxx/${dateformat(yyyy-MM-dd)}/${timestamp()}** is converted to **/opt/ttxx/2017-10-16/1508115701746**.

Time and Date Macro Variables in the Where Clause

Figure 7-8 uses table **SQOOP.CDM_20171016** as an example. The table contains column **DS**, which indicates the time.

Figure 7-8 Table data

SELECT * FROM SQOOP.CDM_20171016

	FOO	BAR	DS
1	5	snap	2017-05-01
2	5	snap	2017-05-01
3	1	google	2017-05-02
4	4	oracle	2017-05-02
5	6	amd	2017-05-02
6	7	nvda	2017-05-02
7	1	google	2017-05-02
8	4	oracle	2017-05-02
9	6	amd	2017-05-02
10	7	nvda	2017-05-02
11	2	facebook	2017-10-15
12	3	tesla	2017-10-15
13	2	facebook	2017-10-15
14	3	tesla	2017-10-15

Suppose that the current date is **2017-10-16** and you want to export data generated the day before the current day (DS = 2017-10-15), then you can set the value of **Where Clause** to **DS='${dateformat(yyyy-MM-dd,-1,DAY)}'** when creating a job. In this way, you can export all data that complies with the DS = 2017-10-15 condition.

Implementing Incremental Synchronization by Configuring the Macro Variables of Date and Time and Scheduled Jobs

Two simple application scenarios are as follows:

- The database table contains column **DS** that indicates the time, the value type of the column is **varchar(30)**, and the inserted time format is similar to **2017-xx-xx**.

  In a scheduled job, the cycle is one day, and the scheduled job is executed at 00:00:00 every day. Set the value of **Where Clause** to **DS='${dateformat(yyyy-MM-dd,-1,DAY)}'**, and then data generated in the previous day will be exported at 00:00:00 every day.

- The database table contains column **time** that indicates the time, the type is **Number**, and the inserted time format is timestamp.

  In a scheduled job, the cycle is one day, and the scheduled job is executed at 00:00:00 every day. Set the value of **Where Clause** to **time between ${timestamp(-1,DAY)} and ${timestamp()}**, and then data generated on the previous day will be exported at 00:00:00 every day.

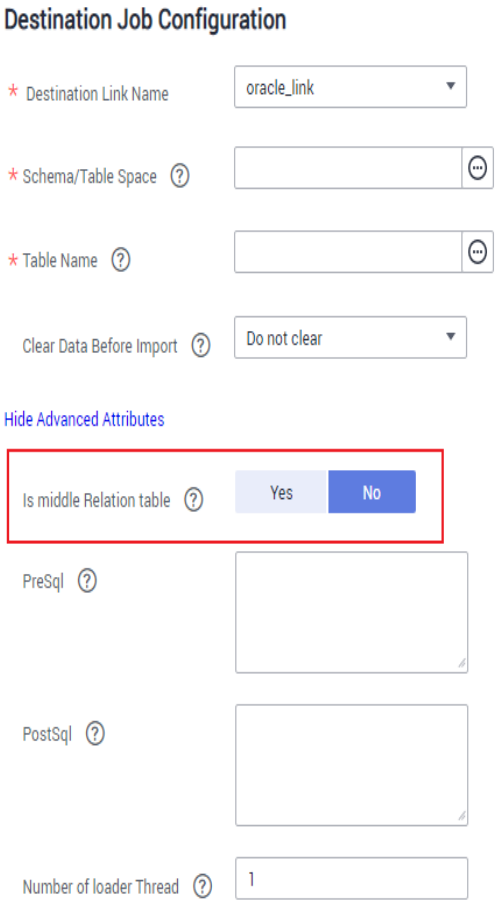
Configuration principles of other application scenarios are the same.

7.3 Migration in Transaction Mode

When a CDM job fails to be executed, CDM rolls back the data to the state before the job starts and automatically deletes data from the destination table.

- Parameter position: When creating a table/file migration job, if the migration source is a relational database, set **Import to Staging Table** in the advanced attributes of **Destination Job Configuration** to determine whether to enable the transaction mode.
- Parameter principle: If you set this parameter to **Yes**, CDM automatically creates a temporary table and imports the data to the temporary table. After the data is imported successfully, CDM migrates the data to the destination table in transaction mode of the database. If the import fails, the destination table is rolled back to the state before the job starts.

Figure 7-9 Migration in transaction mode



NOTE

If you select **Clear part of data** or **Clear all data** for **Clear Data Before Import**, CDM does not roll back the deleted data in transaction mode.

7.4 Encryption and Decryption During File Migration

When you migrate files to a file system, CDM can encrypt and decrypt those files. Currently, CDM supports the following encryption modes:

- **AES-256-GCM**
- **KMS Encryption**

AES-256-GCM

Currently, only AES-256-GCM (NoPadding) is supported. This algorithm is used for encryption at the migration destination and decryption at the migration source. The supported source and destination data sources are as follows:

- Data sources supported by the migration source: HDFS (supported in the binary format)
- Data sources supported by the migration destination: HDFS (supported in the binary format)

The following part describes how to use AES-256-GCM to decrypt the encrypted files to be exported from HDFS and encrypt the files to be imported to HDFS.

- **Configure decryption at the migration source.**

  When you use CDM to create a job for exporting files from HDFS, set the migration source to HDFS and file format to binary, and set the following parameters in the advanced settings of **Source Job Configuration**:

  a. **Encryption**: Select **AES-256-GCM**.

  b. **DEK**: The key must be the same as that configured in encryption. Otherwise, the decrypted data is incorrect and the system does not display an error message.

  c. **IV**: The initialization vector must be the same as that configured in encryption. Otherwise, the decrypted data is incorrect and the system does not display an error message.

  In this way, after CDM exports encrypted files from HDFS, the files written to the migration destination are decrypted plaintext files.

- **Configure encryption at the migration destination.**

  When you create a CDM job to import files to HDFS, set the migration destination to HDFS and file format to binary, and set the following parameters in the advanced settings of **Destination Job Configuration**:

  a. **Encryption**: Select **AES-256-GCM**.

  b. **DEK**: custom encryption key. The key consists of 64 hexadecimal numbers. It is case-insensitive but must contain 64 characters. For example, **DD0AE00DFECD78BF051BCFDA25BD4E320DB0A7AC75A1F3FC3D3C56A457DCDC1B**.

  c. **IV**: custom initialization vector. The initialization vector consists of 32 hexadecimal numbers. It is case-insensitive but must contain 32 characters. For example, **5C91687BA886EDCD12ACBC3FF19A3C3F**.

  In this way, after CDM imports files to HDFS, the files in the destination HDFS are encrypted using the AES-256-GCM algorithm.

KMS Encryption

NOTE

The migration source does not support KMS encryption.

CDM supports KMS encryption if tables, files, or a whole database is migrated to OBS. In the **Advanced Attributes** area of the **Destination Job Configuration** page, set the parameters.

A key must be created in KMS of DEW in advance. For details, see the *Data Encryption Workshop User Guide*.

After KMS encryption is enabled, objects to be uploaded will be encrypted and stored on OBS. When you download the encrypted objects, the encrypted data will be decrypted on the server and displayed in plaintext to users.

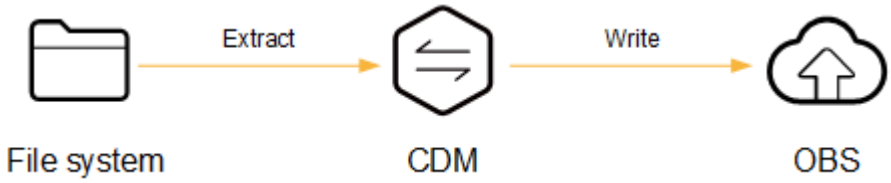
NOTE

- If KMS encryption is enabled, **MD5 verification** cannot be used.
- If the KMS ID of another project is used, change **Project ID** to the ID of the project to which KMS belongs. If KMS and CDM are in the same project, retain the default value of **Project ID**.
- After KMS encryption is performed, the encryption status of the objects on OBS cannot be changed.
- A key in use cannot be deleted. Otherwise, the object encrypted with this key cannot be downloaded.

7.5 MD5 Verification

CDM extracts data from the migration source and writes the data to the migration destination. **Figure 7-10** shows the migration mode when files are migrated to OBS.

Figure 7-10 Migrating files to OBS



During the process, CDM uses MD5 to verify file consistency.

- **Extract**
  - The migration source can be OBS, HDFS, FTP, SFTP, or HTTP. It can check whether the files extracted by CDM are consistent with source files.
  - This function is controlled by the **MD5 File Extension** parameter (available when **File Format** is set to **Binary**) in **Source Job Configuration**. Set this parameter to the file name extension of the MD5 file in the source file system.
  - If a source file **build.sh** and a file for saving MD5 value **build.sh.md5** are located in the same directory, and **MD5 File Extension** is configured, only the file **build.sh.md5** is migrated to the destination. Files without the MD5 value or whose MD5 values do not match fail to be migrated, and the MD5 file is not migrated.
  - If **MD5 File Extension** is not configured, all files are migrated.
- **Write**
  - Currently, this function can be used only when OBS serves as the migration destination. It can check whether the files written to OBS are consistent with those extracted from CDM.
  - This function is controlled by the **Validate MD5 Value** parameter in **Destination Job Configuration**. After the files are read and written to OBS, the MD5 value in the HTTP header is used to verify the files on OBS and the verification result is written to an OBS bucket (the bucket can be the one that does not store migration files). If the migration source does not have the MD5 file, the verification will not be performed.

NOTE

- When files are migrated to a file system, only the extracted files are verified.
- When files are migrated to OBS, both the extracted files and files written to OBS are verified.
- If MD5 verification is used, **KMS encryption** cannot be used.

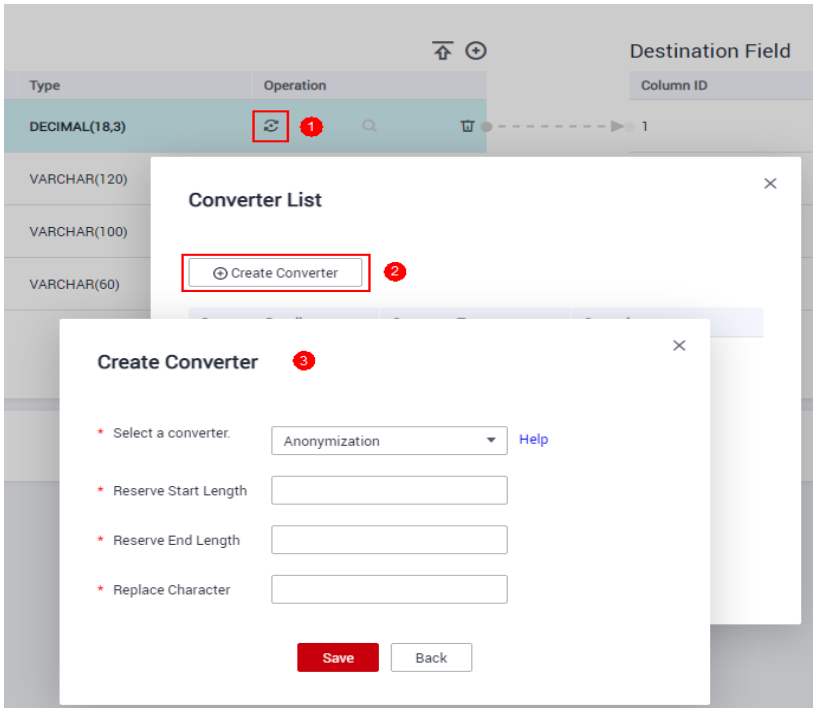
7.6 Configuring Field Converters

Scenario

- After the job parameters are configured, field mapping needs to be configured. You can click in the **Operation** column to create a field converter.
- If files are migrated between FTP, SFTP, OBS, and HDFS and the migration source's **File Format** is set to **Binary**, files will be directly transferred, free from field mapping.

You can create a field converter on the **Map Field** page when creating a table/file migration job.

Figure 7-11 Creating a field converter



CDM can convert fields during migration. Currently, the following field converters are supported:

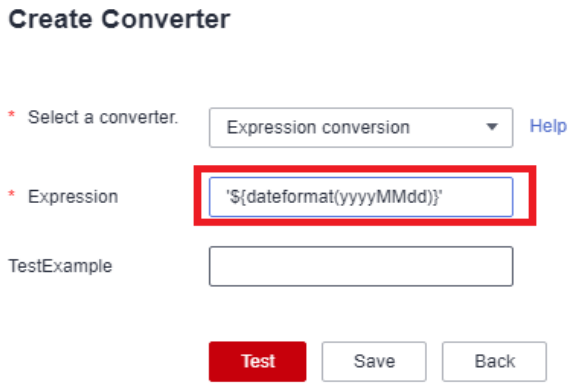
- **Anonymization**
- **Trim**
- **Reverse String**

- **Replace String**
- **Remove line break**
- **Expression Conversion**

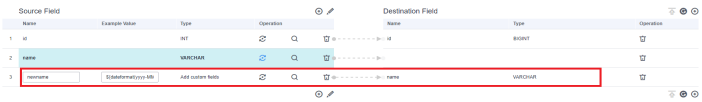
Constraints

- If **Use SQL Statement** is set to **Yes** in the source job configuration, converters cannot be created.
- On the **Map Field** tab page, if CDM fails to obtain all columns by obtaining sample values (for example, when data is exported from HBase, CloudTable, or MongoDB, there is a high probability that CDM failed to obtain all columns), you can click  and select **Add a new field** to add new fields to ensure that the data imported to the migration destination is complete.
- When a relational database, Hive, DLI, or MRS Hudi is used as the migration source, sample values cannot be obtained.
- When SQLServer is the destination, fields of the timestamp type cannot be written. You must change their type (for example, to datetime) so that they can be written.
- Column names are displayed when the source of the migration job is OBS, CSV files are to be migrated, and parameter **Extract first row as columns** is set to **Yes**.
- Field converters configuration is not involved when the binary format is used to migrate files to files.
- In the automatic table creation scenario, you need to manually add fields to the destination table in advance and then add fields to the field mapping.
- After a field is added, its sample value is not displayed on the console. This does not affect the field value transmission. CDM directly writes the field value to the destination end.
- If the field mapping is incorrect, you can adjust the field mapping by dragging fields or clicking  to map fields in batches.
- An expression processes the data of a field. When creating an expression converter, you are not advised to use a time macro. If you need to use a time macro, use either of the following methods (if the source is of the file type, only **Method 1** is supported):
  - Method 1: When creating an expression converter, use two single quotation marks ('') to enclose the expression.

    For example, if expression **${dateformat(yyyy-MM-dd)}** is not enclosed in quotation marks, the hyphen (-) in the value **2017-10-16** parsed from the expression will be recognized as a minus sign, and further calculation will be performed to generate result **1991**, which is incorrect. If you enclose the expression in quotation marks, that is, **'${dateformat(yyyy-MM-dd)}'**, you will obtain **'2017-10-16'**, which is correct.



  - Method 2: Add a custom source field, enter a macro variable of date and time for **Example Value**, and map the field to a destination field again.



- If the data is imported to GaussDB(DWS), you need to select the distribution columns in the destination fields. You are advised to select the distribution columns according to the following rules:
  a. Use the primary key as the distribution column.
  b. If multiple data segments are combined as primary keys, specify all primary keys as the distribution column.
  c. In the scenario where no primary key is available, if no distribution column is selected, DWS uses the first column as the distribution column by default. As a result, data skew risks exist.

Anonymization

This converter is used to hide key information about the character string. For example, if you want to convert **12345678910** to **123****8910**, configure the parameters as follows:

- Set **Reserve Start Length** to **3**.
- Set **Reserve End Length** to **4**.
- Set **Replace Character** to *****.

Trim

This converter is used to automatically delete the spaces before and after a string. No parameters need to be configured.

Reverse String

This converter is used to automatically reverse a string. For example, reverse **ABC** into **CBA**. No parameters need to be configured.

Replace String

This converter is used to replace a character string. You need to configure the object to be replaced and the new value.

Remove line break

This converter is used to delete the newline characters, such as \n, \r, and \r\n from the field.

Expression Conversion

This converter uses the JSP expression language (EL) to convert the current field or a row of data. The JSP EL is used to create arithmetic and logical expressions. Within a JSP EL expression, you can use integers, floating point numbers, strings, the built-in constants **true** and **false** for boolean values, and **null**.

- The expression supports the following environment variables:
  - **value**: indicates the current field value.
  - **row**: indicates the current row, which is an array type.
- The expression supports the following Utils:
  a. If the field is of the string type, convert all character strings into lowercase letters, for example, convert **aBC** to **abc**.

     Expression: StringUtils.lowerCase(value)

  b. Convert all character strings of the current field to uppercase letters.

     Expression: StringUtils.upperCase(value)

  c. Convert the format of the first date field from 2018-01-05 15:15:05 to 20180105.

     Expression: DateUtils.format(DateUtils.parseDate(row[0],"yyyy-MM-dd HH:mm:ss"),"yyyyMMdd")

  d. Convert a timestamp to a date string in *yyyy-MM-dd hh:mm:ss* format, for example, convert **1701312046588** to **2023-11-30 10:40:46**.

     Expression: DateUtils.format(NumberUtils.toLong(value),"yyyy-MM-dd HH:mm:ss")

  e. Convert a date string in the yyyy-MM-dd hh:mm:ss format to a timestamp.

     Expression: DateUtils.getTime(DateUtils.parseDate(value,"yyyy-MM-dd hh:mm:ss"))

  f. If the field value is a date string in *yyyy-MM-dd* format, extract the year from the field value, for example, extract **2017** from **2017-12-01**.

     Expression: StringUtils.substringBefore(value,"-")

  g. If the field value is of the numeric type, convert the value to a new value which is two times greater than the original value:

     Expression: value*2

  h. Convert the field value **true** to **Y** and other field values to **N**.

     Expression: value=="true"?"Y":"N"

  i. If the field value is of the string type and is left empty, convert it to **Default**. Otherwise, the field value will not be converted.

Expression: empty value? "Default":value

j. Convert date format **2018/01/05 15:15:05** to **2018-01-05 15:15:05**:

Expression: DateUtils.format(DateUtils.parseDate(value,"yyyy/MM/dd HH:mm:ss"),"yyyy-MM-dd HH:mm:ss")

k. Obtain a 36-bit universally unique identifier (UUID):

Expression: CommonUtils.randomUUID()

l. If the field is of the string type, capitalize the first letter, for example, convert **cat** to **Cat**.

Expression: StringUtils.capitalize(value)

m. If the field is of the string type, convert the first letter to a lowercase letter, for example, convert **Cat** to **cat**.

Expression: StringUtils.uncapitalize(value)

n. If the field is of the string type, use a space to fill in the character string to the specified length and center the character string. If the length of the character string is not shorter than the specified length, do not convert the character string. For example, convert **ab** to meet the specified length 4.

Expression: StringUtils.center(value,*4*)

o. Delete a newline (including **\n**, **\r**, and **\r\n**) at the end of a character string. For example, convert **abc\r\n\r\n** to **abc\r\n**.

Expression: StringUtils.chomp(value)

p. If the string contains the specified string, **true** is returned; otherwise, **false** is returned. For example, **abc** contains **a** so that **true** is returned.

Expression: StringUtils.contains(value,"*a*")

q. If the string contains any character of the specified string, **true** is returned; otherwise, **false** is returned. For example, **zzabyycdxx** contains either **z** or **a** so that **true** is returned.

Expression: StringUtils.containsAny(value,"*za*")

r. If the string does not contain any one of the specified characters, **true** is returned. If any specified character is contained, **false** is returned. For example, **abz** contains one character of **xyz** so that **false** is returned.

Expression: StringUtils.containsNone(value,"*xyz*")

s. If the string contains only the specified characters, **true** is returned. If any other character is contained, **false** is returned. For example, **abab** contains only characters among **abc** so that **true** is returned.

Expression: StringUtils.containsOnly(value,"*abc*")

t. If the character string is empty or null, convert it to the specified character string. Otherwise, do not convert the character string. For example, convert the empty character string to null.

Expression: StringUtils.defaultIfEmpty(value,*null*)

u. If the string ends with the specified suffix (case sensitive), **true** is returned; otherwise, **false** is returned. For example, if the suffix of **abcdef** is not null, **false** is returned.

Expression: StringUtils.endsWith(value,*null*)

v. If the string is the same as the specified string (case sensitive), **true** is returned; otherwise, **false** is returned. For example, after strings **abc** and **ABC** are compared, **false** is returned.

   Expression: StringUtils.equals(value,"*ABC*")

w. Obtain the first index of the specified character string in a character string. If no index is found, **-1** is returned. For example, the first index of **ab** in **aabaabaa** is 1.

   Expression: StringUtils.indexOf(value,"*ab*")

x. Obtain the last index of the specified character string in a character string. If no index is found, **-1** is returned. For example, the last index of **k** in **aFkyk** is 4.

   Expression: StringUtils.lastIndexOf(value,"*k*")

y. Obtain the first index of the specified character string from the position specified in the character string. If no index is found, **-1** is returned. For example, the first index of **b** obtained after the index 3 of **aabaabaa** is 5.

   Expression: StringUtils.indexOf(value,"*b*",*3*)

z. Obtain the first index of any specified character in a character string. If no index is found, **-1** is returned. For example, the first index of **z** or **a** in **zzabyycdxx.** is 0.

   Expression: StringUtils.indexOfAny(value,"*za*")

aa. If the string contains any Unicode character, **true** is returned; otherwise, **false** is returned. For example, **ab2c** contains only non-Unicode characters so that **false** is returned.

   Expression: StringUtils.isAlpha(value)

ab. If the string contains only Unicode characters and digits, **true** is returned; otherwise, **false** is returned. For example, **ab2c** contains only Unicode characters and digits, so that **true** is returned.

   Expression: StringUtils.isAlphanumeric(value)

ac. If the string contains only Unicode characters, digits, and spaces, **true** is returned; otherwise, **false** is returned. For example, **ab2c** contains only Unicode characters and digits, so that **true** is returned.

   Expression: StringUtils.isAlphanumericSpace(value)

ad. If the string contains only Unicode characters and spaces, **true** is returned; otherwise, **false** is returned. For example, **ab2c** contains Unicode characters and digits so that **false** is returned.

   Expression: StringUtils.isAlphaSpace(value)

ae. If the string contains only printable ASCII characters, **true** is returned; otherwise, **false** is returned. For example, for **!ab-c~**, **true** is returned.

   Expression: StringUtils.isAsciiPrintable(value)

af. If the string is empty or null, **true** is returned; otherwise, **false** is returned.

   Expression: StringUtils.isEmpty(value)

ag. If the string contains only Unicode digits, **true** is returned; otherwise, **false** is returned.

   Expression: StringUtils.isNumeric(value)

ah. Obtain the leftmost characters of the specified length. For example, obtain the leftmost two characters **ab** from **abc**.

Expression: StringUtils.left(value,*2*)

ai. Obtain the rightmost characters of the specified length. For example, obtain the rightmost two characters **bc** from **abc**.

Expression: StringUtils.right(value,*2*)

aj. Concatenate the specified character string to the left of the current character string and specify the length of the concatenated character string. If the length of the current character string is not shorter than the specified length, the character string will not be converted. For example, if **yz** is concatenated to the left of **bat** and the length must be 8 after concatenation, the character string is **yzyzybat** after conversion.

Expression: StringUtils.leftPad(value,*8*,"*yz*")

ak. Concatenate the specified character string to the right of the current character string and specify the length of the concatenated character string. If the length of the current character string is not shorter than the specified length, the character string will not be converted. For example, if **yz** is concatenated to the right of **bat** and the length must be 8 after concatenation, the character string is **batyzyzy** after conversion.

Expression: StringUtils.rightPad(value,*8*,"*yz*")

al. If the field is of the string type, obtain the length of the current character string. If the character string is null, **0** is returned.

Expression: StringUtils.length(value)

am. If the field is of the string type, delete all the specified character strings from it. For example, delete **ue** from **queued** to obtain **qd**.

Expression: StringUtils.remove(value,"*ue*")

an. If the field is of the string type, remove the substring at the end of the field. If the specified substring is not at the end of the field, no conversion is performed. For example, remove **.com** at the end of **www.domain.com**.

Expression: StringUtils.removeEnd(value,"*.com*")

ao. If the field is of the string type, delete the substring at the beginning of the field. If the specified substring is not at the beginning of the field, no conversion is performed. For example, delete **www.** at the beginning of **www.domain.com**.

Expression: StringUtils.removeStart(value,"*www.*")

ap. If the field is of the string type, replace all the specified character strings in the field. For example, replace **a** in **aba** with **z** to obtain **zbz**.

Expression: StringUtils.replace(value,"*a*","*z*")

aq. If the field is of the string type, replace multiple characters in the character string at a time. For example, replace **h** in **hello** with **j** and **o** with **y** to obtain **jelly**.

Expression: StringUtils.replaceChars(value,"*ho*","*jy*")

ar. If the string starts with the specified prefix (case sensitive), **true** is returned; otherwise, **false** is returned. For example, **abcdef** starts with **abc**, so that **true** is returned.

Expression: StringUtils.startsWith(value,"*abc*")

as. If the field is of the string type, delete all the specified characters at the beginning and end of the field. the field. For example, delete all **x**, **y**, **z**, and **b** from **abcyx** to obtain **abc**.

Expression: StringUtils.strip(value,"*xyz*b")

at. If the field is of the string type, delete all the specified characters at the end of the field, for example, delete the **abc** string at the end of the field.

Expression: StringUtils.stripEnd(value,*"abc"*)

au. If the field is of the string type, delete all the specified characters at the beginning of the field, for example, delete all spaces at the beginning of the field.

Expression: StringUtils.stripStart(value,*null*)

av. If the field is of the string type, obtain the substring after the specified position (the index starts from 0, including the character at the specified position) of the character string. If the specified position is a negative number, calculate the position in the descending order. The first digit at the end is -1. For example, obtain the second character (c) of **abcde** and the string after it, that is, **cde**.

Expression: StringUtils.substring(value,*2*)

aw. If the field is of the string type, obtain the substring in a specified range (the index starts from 0, including the character at the start and excluding the character at the end). If the range is a negative number, calculate the position in the descending order. The first digit at the end is -1. For example, obtain the string between the second character (c) and fourth character (e) of **abcde**, that is, **cd**.

Expression: StringUtils.substring(value,*2*,4)

ax. If the field is of the string type, obtain the substring after the first specified character. For example, obtain the substring after the first **b** in **abcba**, that is, **cba**.

Expression: StringUtils.substringAfter(value,"*b*")

ay. If the field is of the string type, obtain the substring after the last specified character. For example, obtain the substring after the last **b** in **abcba**, that is, **a**.

Expression: StringUtils.substringAfterLast(value,"*b*")

az. If the field is of the string type, obtain the substring before the first specified character. For example, obtain the substring before the first **b** in **abcba**, that is, **a**.

Expression: StringUtils.substringBefore(value,"*b*")

ba. If the field is of the string type, obtain the substring before the last specified character. For example, obtain the substring before the last **b** in **abcba**, that is, **abc**.

Expression: StringUtils.substringBeforeLast(value,"*b*")

bb. If the field is of the string type, obtain the substring nested within the specified string. If no substring is found, **null** is returned. For example, obtain the substring between **tag** in **tagabctag**, that is, **abc**.

Expression: StringUtils.substringBetween(value,"*tag*")

bc. If the field is of the string type, delete the control characters (char≤32) at both ends of the character string, for example, delete the spaces at both ends of the character string.

Expression: StringUtils.trim(value)

bd. Convert the character string to a value of the byte type. If the conversion fails, **0** is returned.

Expression: NumberUtils.toByte(value)

be. Convert the character string to a value of the byte type. If the conversion fails, the specified value, for example, **1**, is returned.

Expression: NumberUtils.toByte(value,*1*)

bf. Convert the character string to a value of the double type. If the conversion fails, **0.0d** is returned.

Expression: NumberUtils.toDouble(value)

bg. Convert the character string to a value of the double type. If the conversion fails, the specified value, for example, **1.1d**, is returned.

Expression: NumberUtils.toDouble(value,*1.1d*)

bh. Convert the character string to a value of the float type. If the conversion fails, **0.0f** is returned.

Expression: NumberUtils.toFloat(value)

bi. Convert the character string to a value of the float type. If the conversion fails, the specified value, for example, **1.1f**, is returned.

Expression: NumberUtils.toFloat(value,*1.1f*)

bj. Convert the character string to a value of the int type. If the conversion fails, **0** is returned.

Expression: NumberUtils.toInt(value)

bk. Convert the character string to a value of the int type. If the conversion fails, the specified value, for example, **1**, is returned.

Expression: NumberUtils.toInt(value,*1*)

bl. Convert the character string to a value of the long type. If the conversion fails, **0** is returned.

Expression: NumberUtils.toLong(value)

bm. Convert the character string to a value of the long type. If the conversion fails, the specified value, for example, **1L**, is returned.

Expression: NumberUtils.toLong(value,*1L*)

bn. Convert the character string to a value of the short type. If the conversion fails, **0** is returned.

Expression: NumberUtils.toShort(value)

bo. Convert the character string to a value of the short type. If the conversion fails, the specified value, for example, **1**, is returned.

Expression: NumberUtils.toShort(value,*1*)

bp. Convert the IP string to a value of the long type, for example, convert **10.78.124.0** to **172915712**.

Expression: CommonUtils.ipToLong(value)

bq. Read an IP address and physical address mapping file from the network, and download the mapping file to the map collection. *url* indicates the

address for storing the IP mapping file, for example, **http://10.114.205.45:21203/sqoop/IpList.csv**.

Expression: HttpsUtils.downloadMap("*url*")

br. Cache the IP address and physical address mappings and specify a key for retrieval, for example, **ipList**.

Expression: CommonUtils.setCache("*ipList*",HttpsUtils.downloadMap("*url*"))

bs. Obtain the cached IP address and physical address mappings.

Expression: CommonUtils.getCache("*ipList*")

bt. Check whether the IP address and physical address mappings are cached.

Expression: CommonUtils.cacheExists("*ipList*")

bu. Based on the specified offset type (month/day/hour/minute/second) and offset (positive number indicates increase and negative number indicates decrease), convert the time in the specified format to a new time, for example, add 8 hours to **2019-05-21 12:00:00**.

Expression: DateUtils.getCurrentTimeByZone("*yyyy-MM-dd HH:mm:ss*",value, "*hour*", *8*)

bv. If the value is empty or null, "aaa" is returned. Otherwise, **value** is returned.

Expression: StringUtils.defaultIfEmpty(value, *"aaa"*)

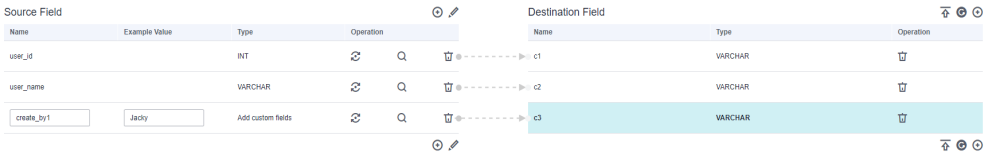
7.7 Adding Fields

Scenario

- After job parameters are configured, field mapping needs to be configured. You can customize new fields by clicking on the **Map Field** page.
- If files are migrated between FTP, SFTP, OBS, and HDFS and the migration source's **File Format** is set to **Binary**, files will be directly transferred, free from field mapping.
- In other scenarios, CDM automatically maps fields of the source table and the destination table. You need to check whether the mapping and time format are correct. For example, check whether the source field type can be converted into the destination field type.

You can click on the **Map Field** page and select **Add** to customize a new field. This field is usually used to mark the database source to ensure the integrity of the data imported to the migration destination.

Figure 7-12 Field mapping



Currently, the following field types are supported:

- **Constant Parameter**

  Constant parameters are fixed parameters and do not need to be reconfigured. For example, **lable** = **friends** is used to identify a constant value.

- **Variables**

  You can use variables such as time macros, table name macros, and version macros to mark database source information. The variable syntax is ${variable}, where **variable** indicates a variable. For example, **input_time** = **${timestamp()}** indicates the timestamp of the current time.

- Expression

  You can use the expression language to dynamically generate parameter values based on the running environment. The expression syntax is #{expr}, where **expr** indicates an expression. For example, **time** = **#{DateUtil.now()}** is used to identify the current date string.

Constraints

- On the **Map Field** tab page, if CDM fails to obtain all columns by obtaining sample values (for example, when data is exported from HBase, CloudTable, or MongoDB, there is a high probability that CDM failed to obtain all columns), you can click  and select **Add a new field** to add new fields to ensure that the data imported to the migration destination is complete.
- When a relational database, Hive, DLI, or MRS Hudi is used as the migration source, sample values cannot be obtained.
- When SQLServer is the destination, fields of the timestamp type cannot be written. You must change their type (for example, to datetime) so that they can be written.
- Column names are displayed when the source of the migration job is OBS, CSV files are to be migrated, and parameter **Extract first row as columns** is set to **Yes**.
- Field mapping is not involved when the binary format is used to migrate files to files.
- In the automatic table creation scenario, you need to manually add fields to the destination table in advance and then add fields to the field mapping.
- After a field is added, its sample value is not displayed on the console. This does not affect the field value transmission. CDM directly writes the field value to the destination end.
- If the field mapping is incorrect, you can adjust the field mapping by dragging fields or clicking  to map fields in batches.
- If the data is imported to DWS, you need to select the distribution columns in the destination fields. You are advised to select the distribution columns according to the following principles:

  a. Use the primary key as the distribution column.

  b. If multiple data segments are combined as primary keys, specify all primary keys as the distribution column.

  c. In the scenario where no primary key is available, if no distribution column is selected, DWS uses the first column as the distribution column by default. As a result, data skew risks exist.

- If a source field type is not supported, convert the field type to a type supported by CDM by referring to **Converting Unsupported Data Types**.

7.8 Migrating Files with Specified Names

You can migrate files (a maximum of 50) with specified names from FTP, OBS, or SFTP at a time. The exported files can only be written to the same directory on the migration destination.

When creating a table/file migration job, if the migration source is FTP, OBS, or SFTP, **Source Directory/File** can contain a maximum of 50 file names, which are separated by vertical bars (|). You can also customize a file separator.

NOTE

1. CDM supports incremental file migration (by skipping repeated files), but does not support resumable transfer.

   For example, if three files are to be migrated and the second file fails to be migrated due to the network fault. When the migration task is started again, the first file is skipped. The second file, however, cannot be migrated from the point where the fault occurs, but can only be migrated again.
2. During file migration, a single task supports millions of files. If there are too many files in the directory to be migrated, you are advised to split the files into different directories and create multiple tasks.

7.9 Regular Expressions for Separating Semi-structured Text

During table/file migration, CDM uses delimiters to separate fields in CSV files. However, delimiters cannot be used in complex semi-structured data because the field values also contain delimiters. In this case, the regular expression can be used to separate the fields.

The regular expression is configured in **Source Job Configuration**. The migration source must be an object storage or file system, and **File Format** must be **CSV**.

During the migration of CSV files, CDM can use regular expressions to separate fields and write parsed results to the migration destination. For details about the syntax of the regular expression, refer to the related documents. This section describes the regular expressions of the following log files:

- **Log4J Log**
- **Log4J Audit Log**
- **Tomcat Log**
- **Django Log**
- **Apache Server Log**

Log4J Log

- Log sample:

```
2018-01-11 08:50:59,001 INFO
[org.apache.sqoop.core.SqoopConfiguration.configureClassLoader(SqoopConfiguration.java:251)]
Adding jars to current classloader from property: org.apache.sqoop.classpath.extra
```


- Regular expression:
  ```
  ^(\d.*\d) (\w*)  \[(.*)\] (\w.*).*
  ```
- Parsing result:

Table 7-2 Log4J log parsing result

Column Number	Example Value
1	2018-01-11 08:50:59,001
2	INFO
3	org.apache.sqoop.core.SqoopConfiguration.configureClassLoader(SqoopConfiguration.java:251)
4	Adding jars to current classloader from property: org.apache.sqoop.classpath.extra

Log4J Audit Log

- Log sample:
  ```
  2018-01-11 08:51:06,156 INFO
  [org.apache.sqoop.audit.FileAuditLogger.logAuditEvent(FileAuditLogger.java:61)]
  user=sqoop.anonymous.user    ip=189.xxx.xxx.75    op=show    obj=version    objId=x
  ```
- Regular expression:
  ```
  ^(\d.*\d) (\w*)  \[(.*)\] user=(\w.*)    ip=(\w.*)    op=(\w.*)    obj=(\w.*)    objId=(.*).*
  ```
- Parsing result:

Table 7-3 Log4J audit log parsing result

Column Number	Example Value
1	2018-01-11 08:51:06,156
2	INFO
3	org.apache.sqoop.audit.FileAuditLogger.logAuditEvent(FileAuditLogger.java:61)
4	sqoop.anonymous.user
5	189.xxx.xxx.75
6	show
7	version
8	x

Tomcat Log

- Log sample:
  ```
  11-Jan-2018 09:00:06.907 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Name:               Linux
  ```
- Regular expression:
  ```
  ^(\d.*\d) (\w*) \[(.*)\] ([\w\.]*) (\w.*).*
  ```
- Parsing result:

  Table 7-4 Tomcat log parsing result

Column Number	Example Value
1	11-Jan-2018 09:00:06.907
2	INFO
3	main
4	org.apache.catalina.startup.VersionLoggerListener.log
5	OS Name:Linux

Django Log

- Log sample:
  ```
  [08/Jan/2018 20:59:07 ] settings     INFO     Welcome to Hue 3.9.0
  ```
- Regular expression:
  ```
  ^\[(.*)\] (\w*)     (\w*)     (.*).*
  ```
- Parsing result:

  Table 7-5 Django log parsing result

Column Number	Example Value
1	08/Jan/2018 20:59:07
2	settings
3	INFO
4	Welcome to Hue 3.9.0

Apache Server Log

- Log sample:
  ```
  [Mon Jan 08 20:43:51.854334 2018] [mpm_event:notice] [pid 36465:tid 140557517657856] AH00489: Apache/2.4.12 (Unix) OpenSSL/1.0.1t configured -- resuming normal operations
  ```


- Regular expression:
  ```
  ^\[(.*)\] \[(.*)\] \[(.*)\] (.*).*
  ```
- Parsing result:

Table 7-6 Apache server log parsing result

Column Number	Example Value
1	Mon Jan 08 20:43:51.854334 2018
2	mpm_event:notice
3	pid 36465:tid 140557517657856
4	AH00489: Apache/2.4.12 (Unix) OpenSSL/1.0.1t configured -- resuming normal operations

7.10 Recording the Time When Data Is Written to the Database

When you create a job on the CDM console to migrate tables or files of a relational database, you can add a field to record the time when they were written to the database.

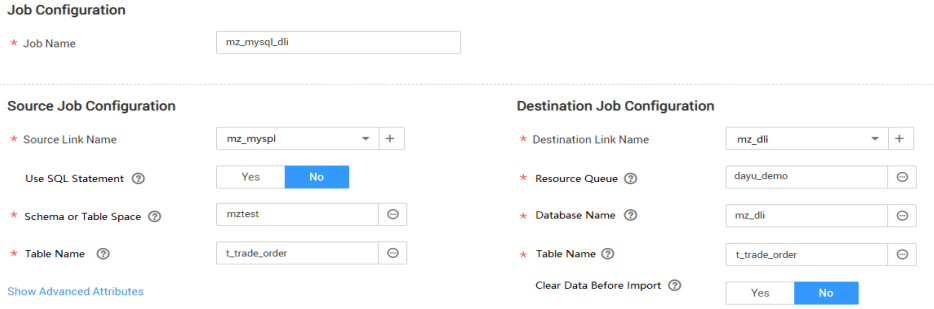
Prerequisites

- A link has been created, and the source end of the connector is a relational database.
- The destination data table contains a date and time field or timestamp field. In the automatic table creation scenario, you need to manually create the date and time field or timestamp field in the destination table in advance.

Creating a Table/File Migration Job

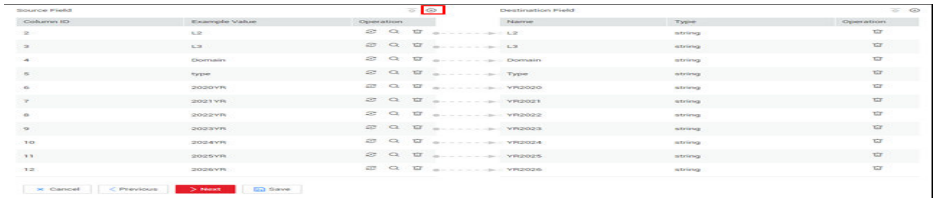
Step 1 Create a table/file migration job, and select the created source connector and destination connector.

Figure 7-13 Configuring the job



Step 2 Click **Next** to go to the **Map Field** page and click ⊕.

Figure 7-14 Configuring field mapping



Step 3 Click the **Custom Fields** tab, set the field name and value, and click **OK**.

Name: Enter **InputTime**.

Value: Enter **${timestamp()}**. For more time macro variables, see **Table 7-7**.

Figure 7-15 Add Field

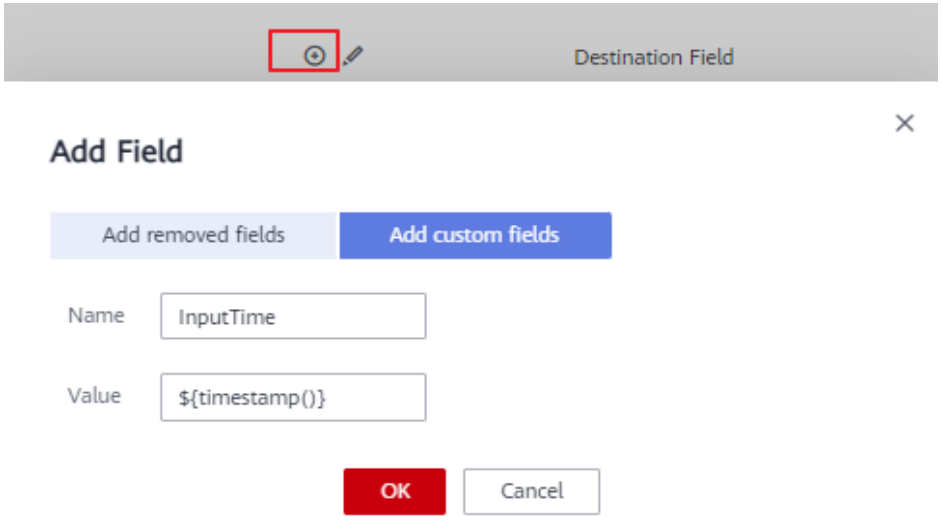


Table 7-7 Macro variable definition of time and date

Macro Variable	Description	Display Effect
${dateformat(yyyy-MM-dd)}	Returns the current date in **yyyy-MM-dd** format.	2017-10-16
${dateformat(yyyy/MM/dd)}	Returns the current date in **yyyy/MM/dd** format.	2017/10/16
${dateformat(yyyy_MM_dd HH:mm:ss)}	Returns the current time in **yyyy_MM_dd HH:mm:ss** format.	2017_10_16 09:00:00
${dateformat(yyyy-MM-dd HH:mm:ss, -1, DAY)}	Returns the current time in **yyyy-MM-dd HH:mm:ss** format. The date is one day before the current day.	2017-10-15 09:00:00

Macro Variable	Description	Display Effect
${timestamp()}	Returns the timestamp of the current time, that is, the number of milliseconds that have elapsed since 00:00:00 on January 1, 1970.	1508115600000
${timestamp(-10, MINUTE)}	Returns the timestamp generated 10 minutes before the current time point.	1508115000000
${timestamp(dateformat(yyyyMMdd))}	Returns the timestamp of 00:00:00 of the current day.	1508083200000
${timestamp(dateformat(yyyyMMdd,-1,DAY))}	Returns the timestamp of 00:00:00 of the previous day.	1507996800000
${timestamp(dateformat(yyyyMMddHH))}	Returns the timestamp of the current hour.	1508115600000

NOTE

- After a field is added, its sample value is not displayed on the console. This does not affect the field value transmission. CDM directly writes the field value to the destination end.
- The **Custom Fields** tab is available only when the source connector is JDBC, HBase, MongoDB, Elasticsearch, or Kafka, or the destination connector is HBase.
- After adding the fields, ensure that the customized import time field matches the field type of the destination table.

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

Step 5 Click **Save and Run**. On the **Table/File Migration** page, you can view the job execution progress and result.

Step 6 After the job is successfully executed, in the **Operation** column of the job, click **Historical Record** to view the job's historical execution records and read/write statistics.

On the **Historical Record** page, click **Log** to view the job logs.

Step 7 Go to the destination data source to check the time when the data is imported to the database.

----End

7.11 File Formats

When creating a CDM job, you need to specify **File Format** in the job parameters of the migration source and destination in some scenarios. This section describes

the application scenarios, subparameters, common parameters, and usage examples of the supported file formats.

- **CSV**
- **JSON**
- **Binary**
- **Common parameters**
- **Solutions to File Format Problems**

CSV

To read or write a CSV file, set **File Format** to **CSV**. The CSV format can be used in the following scenarios:

- Import files to a database or NoSQL.
- Export data from a database or NoSQL to files.

After selecting the CSV format, you can also configure the following optional sub-parameters:

1. Line Separator

2. Field Delimiter

3. Encoding Type

4. Use Quote Character

5. Use RE to Separate Fields

6. Use First Row as Header

7. File Size

1. **Line Separator**

   Character used to separate lines in a CSV file. The value can be a single character, multiple characters, or special characters. Special characters can be entered using the URL encoded characters. The following table lists the URL encoded characters of commonly used special characters.

   Table 7-8 URL encoded characters of special characters

Special Character	URL Encoded Character
Space	%20
Tab	%09
%	%25
Enter	%0d
Newline character	%0a
Start of heading\u0001 (SOH)	%01

2. **Field Delimiter**

   Character used to separate columns in a CSV file. The value can be a single character, multiple characters, or special characters. For details, see **Table 7-8**.

3. **Encoding Type**

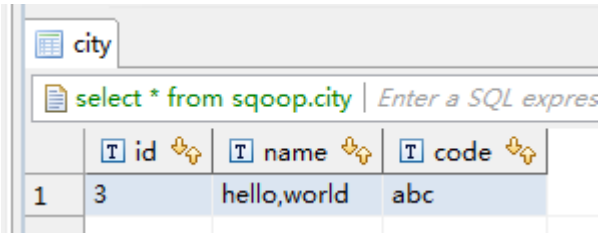
   Encoding type of a CSV file. The default value is **UTF-8**. Some Chinese characters are encoded by GBK.

   If this parameter is specified at the migration source, the specified encoding type is used to parse the file. If this parameter is specified at the migration destination, the specified encoding type is used to write data to the file.

4. **Use Quote Character**

   - Exporting data from a database or NoSQL to CSV files (configuring **Use Quote Character** at the migration destination): If a field delimiter appears in the character string of a column of data at the migration source, set **Use Quote Character** to **Yes** at the migration destination to quote the character string as a whole and write it into the CSV file. Currently, CDM uses double quotation marks ("") as the quote character only. **Figure 7-16** shows that the value of the **name** field in the database contains a comma (,).

     Figure 7-16 Field value containing the field delimiter

     If you do not use the quote character, the exported CSV file is displayed as follows:

     ```
     3.hello,world,abc
     ```

     If you use the quote character, the exported CSV file is displayed as follows:

     ```
     3,"hello,world",abc
     ```

     If the data in the database contains double quotation marks ("") and you set **Use Quote Character** to **Yes**, the quote character in the exported CSV file is displayed as three double quotation marks (""""). For example, if the value of a field is **a"hello,world"c**, the exported data is as follows:

     ```
     """a"hello,world"c"""
     ```

   - Exporting CSV files to a database or NoSQL (configuring **Use Quote Character** at the migration source): If you want to import the CSV files with quoted values to a database correctly, set **Use Quote Character** to **Yes** at the migration source to write the quoted values as a whole.

5. **Use RE to Separate Fields**

   This function is used to parse complex semi-structured text, such as log files. For details, see **Using Regular Expressions to Separate Semi-structured Text**.

6. **Use First Row as Header**

This parameter is used when CSV files are exported to other locations. If this parameter is specified at the migration source, CDM uses the first row as the header when extracting data. When the CSV files are transferred, the headers are skipped. The number of rows extracted from the migration source is more than the number of rows written to the migration destination. The log files will output the information that the header is skipped during the migration.

7. **File Size**

   This parameter is used when data is exported from the database to a CSV file. If a table contains a large amount of data, a large CSV file is generated after migration, which is inconvenient to download or view. In this case, you can specify this parameter at the migration destination so that multiple CSV files with the specified size can be generated. The value of this parameter is an integer. The unit is MB.

JSON

The following describes information about the JSON format:

- **JSON Types Supported by CDM**
- **JSON Reference Node**
- **Copying Data from a JSON File**

1. **JSON types supported by CDM: JSON object and JSON array**
   - JSON object: A JSON file contains a single object or multiple objects separated/merged by rows.
     i. The following is a single JSON object:

     ```
     {
         "took" : 190,
         "timed_out" : false,
         "total" : 1000001,
         "max_score" : 1.0
     }
     ```

     ii. The following are JSON objects separated by rows:

     ```
     {"took" : 188, "timed_out" : false, "total" : 1000003, "max_score" : 1.0 }
     {"took" : 189, "timed_out" : false, "total" : 1000004, "max_score" : 1.0 }
     ```

     iii. The following are merged JSON objects:

     ```
     {
        "took": 190,
        "timed_out": false,
        "total": 1000001,
        "max_score": 1.0
     }
     {
        "took": 191,
        "timed_out": false,
        "total": 1000002,
        "max_score": 1.0
     }
     ```

   - JSON array: A JSON file is a JSON array consisting of multiple JSON objects.

     ```
     [{
         "took" : 190,
         "timed_out" : false,
         "total" : 1000001,
         "max_score" : 1.0
     },
     {
         "took" : 191,
     ```



```
    "timed_out" : false,
    "total" : 1000001,
    "max_score" : 1.0
 }]
```

2. **JSON Reference Node**

   Root node that records data. The data corresponding to the node is a JSON array. CDM extracts data from the array in the same mode. Use periods (.) to separate multi-layer nested JSON nodes.

3. **Copying Data from a JSON File**

   a. Example 1: Extract data from multiple objects that are separated or merged. A JSON file contains multiple JSON objects. The following gives an example:

```
{
  "took": 190,
  "timed_out": false,
  "total": 1000001,
  "max_score": 1.0
}
{
  "took": 191,
  "timed_out": false,
  "total": 1000002,
  "max_score": 1.0
}
{
  "took": 192,
  "timed_out": false,
  "total": 1000003,
  "max_score": 1.0
}
```

   To extract data from the JSON object and write data to the database in the following formats, set **File Format** to **JSON** and **JSON Type** to **JSON object**, and then map fields.

took	timedOut	total	maxScore
190	false	1000001	1.0
191	false	1000002	1.0
192	false	1000003	1.0

   b. Example 2: Extract data from the reference node. A JSON file contains a single JSON object, but the valid data is on a data node. The following gives an example:

```
{
  "took": 190,
  "timed_out": false,
  "hits": {
     "total": 1000001,
     "max_score": 1.0,
     "hits":
      [{
        "_id": "650612",
        "_source": {
            "name": "tom",
            "books": ["book1","book2","book3"]
        }
      },
      {
         "_id": "650616",
```



```
            "_source": {
                "name": "tom",
                "books": ["book1","book2","book3"]
            }
        },
        {
            "_id": "650618",
            "_source": {
                "name": "tom",
                "books": ["book1","book2","book3"]
            }
        }]
    }
}
```

To write data to the database in the following formats, set **File Format** to **JSON**, **JSON Type** to **JSON object**, and **JSON Reference Node** to **hits.hits**, and then map fields.

ID	SourceName	SourceBooks
650612	tom	["book1","book2","book3"]
650616	tom	["book1","book2","book3"]
650618	tom	["book1","book2","book3"]

c. Example 3: Extract data from the JSON array. A JSON file is a JSON array consisting of multiple JSON objects. The following gives an example:

```
[{
    "took" : 190,
    "timed_out" : false,
    "total" : 1000001,
    "max_score" : 1.0
},
{
    "took" : 191,
    "timed_out" : false,
    "total" : 1000002,
    "max_score" : 1.0
}]
```

To write data to the database in the following formats, set **File Format** to **JSON** and **JSON Type** to **JSON array**, and then map fields.

took	timedOut	total	maxScore
190	false	1000001	1.0
191	false	1000002	1.0

d. Example 4: Configure a converter when parsing the JSON file. On the premise of **example 2**, to add the **hits.max_score** field to all records, that is, to write the data to the database in the following formats, perform the following operations:

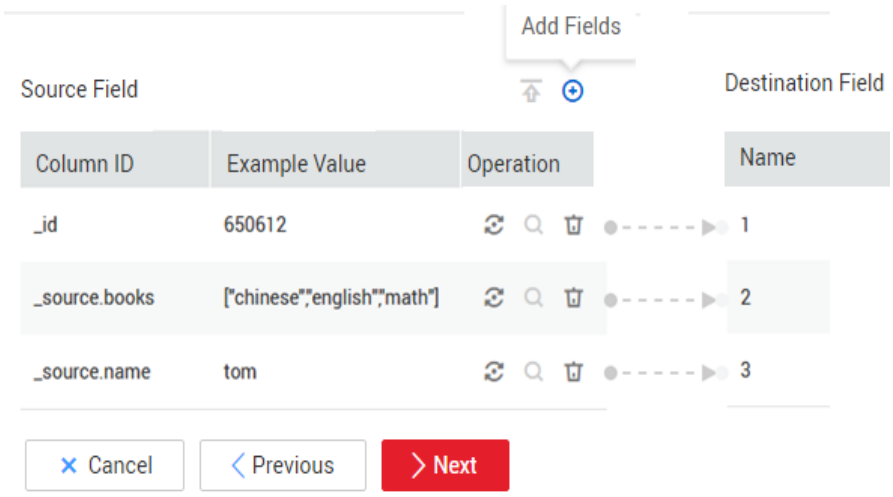
ID	SourceName	SourceBooks	MaxScore
650612	tom	["book1","book2","book3"]	1.0

ID	SourceName	SourceBooks	MaxScore
650616	tom	["book1","book2","book3"]	1.0
650618	tom	["book1","book2","book3"]	1.0

Set **File Format** to **JSON**, **JSON Type** to **JSON object**, and **JSON Reference Node** to **hits.hits**, and then create a converter.

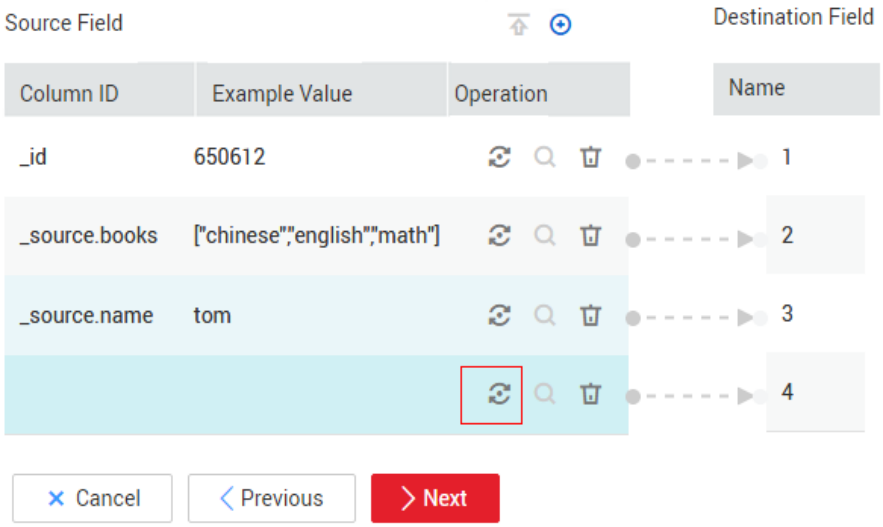
i. Click ⊕ to add a field.

Figure 7-17 Adding a field



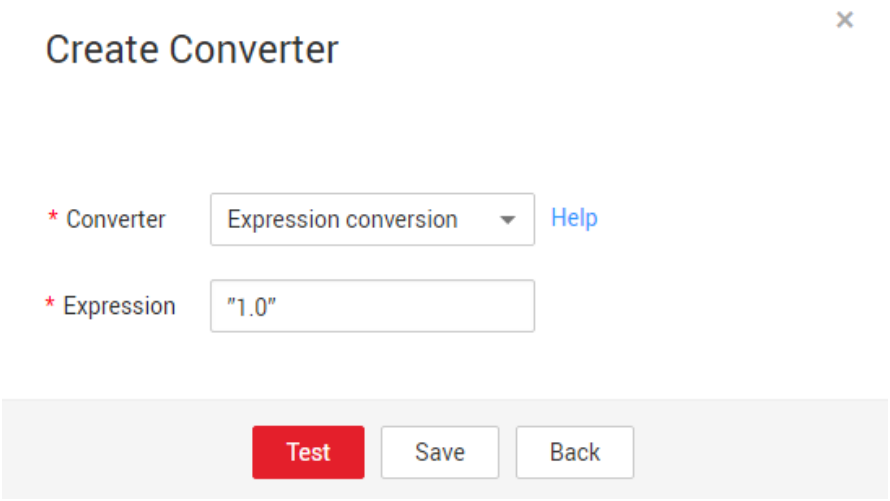
ii. Click ⟳ to create a converter for the new field.

Figure 7-18 Creating a field converter



iii. Set **Converter** to **Expression conversion**, enter **"1.0"** in the **Expression** text box, and click **Save**.

Figure 7-19 Configuring a field converter



Binary

If you want to copy files between file systems, you can select the binary format. Files can be transferred in binary format at a high speed and stable performance. In addition, field mapping is not required in the second step of the job.

- **Directory structure for file transfer**

  CDM can transfer a single file or all files in a directory at a time. After the files are transferred to the migration destination, the directory structure remains unchanged.

- **Migrating incremental files**

  When you use CDM to transfer files in binary format, configure **Duplicate File Processing Method** at the migration destination for incremental file migration. For details, see **Incremental File Migration**.

  During incremental file migration, set **Duplicate File Processing Method** to **Skip**. If new files exist at the migration source or a failure occurs during the migration, run the job again, so that the migrated files will not be migrated repeatedly.

- **Write to Temporary File**

  When migrating files in binary format, you can specify whether to write the files to a temporary file at the migration destination. If this parameter is specified, the file is written to a temporary file during file replication. After the file is successfully migrated, run the **rename** or **move** command to restore the file at the migration destination.

- **Generate MD5 Hash Value**

  An MD5 hash value is generated for each transferred file, and the value is recorded in a new **.md5** file. You can specify the directory where the MD5 value is generated.

Common parameters

- **Start Job by Marker File**

  In automation scenarios, a scheduled task is configured on CDM to periodically read files from the migration source. However, files are being generated at the migration source. As a result, CDM reads data repeatedly or fails to read data from the migration source. You can specify the marker file for starting a job as **ok.txt** in the job parameters of the migration source. After the file is successfully generated at the migration source, the **ok.txt** file is generated in the file directory. In this way, CDM can read the complete file.

  In addition, you can set the suspension period. Within the suspension period, CDM periodically queries whether the marker file exists. If the file does not exist after the suspension period expires, the job fails.

  The marker file will not be migrated.

- **Job Success Marker File**

  After data is successfully migrated to a file system, an empty file is generated in the destination directory. You can specify the file name. Generally, this parameter is used together with **Start Job by Marker File**.

  Note that the file cannot be confused with the file to be transferred. For example, if the file to be transferred is **finish.txt** and the job success marker file is set to **finish.txt**, the two files will overwrite each other.

- **Filter**

  When using CDM to migrate files, you can specify a filter to filter files. Files can be filtered by wildcard character or time filter.

  - If you select **Wildcard**, CDM migrates only the paths or files that meet the filter condition.
  - If you select **Time Filter**, CDM migrates only the files modified after the specified time point.

  For example, the **/table/** directory stores a large number of data table directories divided by day. **DRIVING_BEHAVIOR_20180101** to **DRIVING_BEHAVIOR_20180630** store all data of **DRIVING_BEHAVIOR** from January to June. To migrate only the table data of **DRIVING_BEHAVIOR** in March, set **Source Directory/File** to **/table**, **Filter Type** to **Wildcard**, and **Path Filter** to **DRIVING_BEHAVIOR_201803***.

Solutions to File Format Problems

1. When data in a database is exported to a CSV file, if the data contains commas (,), the data in the exported CSV file is disordered.

   The following solutions are available:

   a. Specify a field delimiter.

      Use a character that does not exist in the database or a rare non-printable character as the field delimiter. For example, set **Field Delimiter** at the migration destination to **%01**. In this way, the exported field delimiter is **\u0001**. For details, see **Table 7-8**.

   b. Use the quote character.

      Set **Use Quote Character** to **Yes** at the migration destination. In this way, if the field in the database contains the field delimiter, CDM quotes the

field using the quote character and write the field as a whole to the CSV file.

2. The data in the database contains line separators.

   Scenario: When you use CDM to export a table in the MySQL database (a field value contains the line separator **\n**) to a CSV file, and then use CDM to import the exported CSV file to MRS HBase, data in the exported CSV file is truncated.

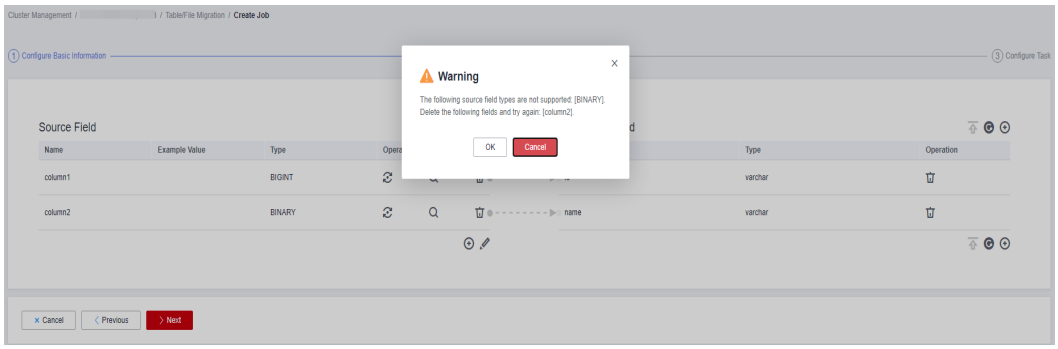
   Solution: Specify a line separator.

   When you use CDM to export MySQL table data to a CSV file, set **Line Separator** at the migration destination to **%01** (ensure that the value does not appear in the field value). In this way, the line separator in the exported CSV file is **%01**. Then use CDM to import the CSV file to MRS HBase. Set **Line Separator** at the migration source to **%01**. This avoids data truncation.

7.12 Converting Unsupported Data Types

Scenario

When field mapping is configured on CDM, a message is displayed indicating that the data type of the field is not supported and the field needs to be deleted. If you need to use this field, you can use SQL statements to convert the field type in the source job configuration to the type supported by CDM for data migration.



Procedure

Step 1 Modify the CDM migration job and enable **Use SQL Statement**.

NOTE

The SQL statement format is as follows: **select id,cast**(*Original field name* **as INT) as** *New field name, which can be the same as the original field name* **from schemaName.tableName;**

For example, select `id`, `name`, cast(`sex` AS char(255)) AS `sex` from `test_1117869`.`test_no_support_type`;

Step 2 Wait for the fields to be converted to the data types supported by CDM.

----End

8 Tutorials

8.1 Creating an MRS Hive Link

MRS Hive links are applicable to the MapReduce Service (MRS). This tutorial describes how to create an MRS Hive link.

Prerequisites

- You have created a CDM cluster.
- You have obtained the Manager IP address, and administrator account and password of the MRS cluster, and the account has the permissions to import and export data.
- The MRS cluster and the CDM cluster can communicate with each other. The following requirements must be met for network interconnection:
  - 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.
  - 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 they are in the same VPC but in different subnets or security groups, you must configure routing rules and security group rules. For details about how to configure routing rules, see **configuring routes**. For details about how to configure security group rules, see **configuring security group rules**.
  - 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.

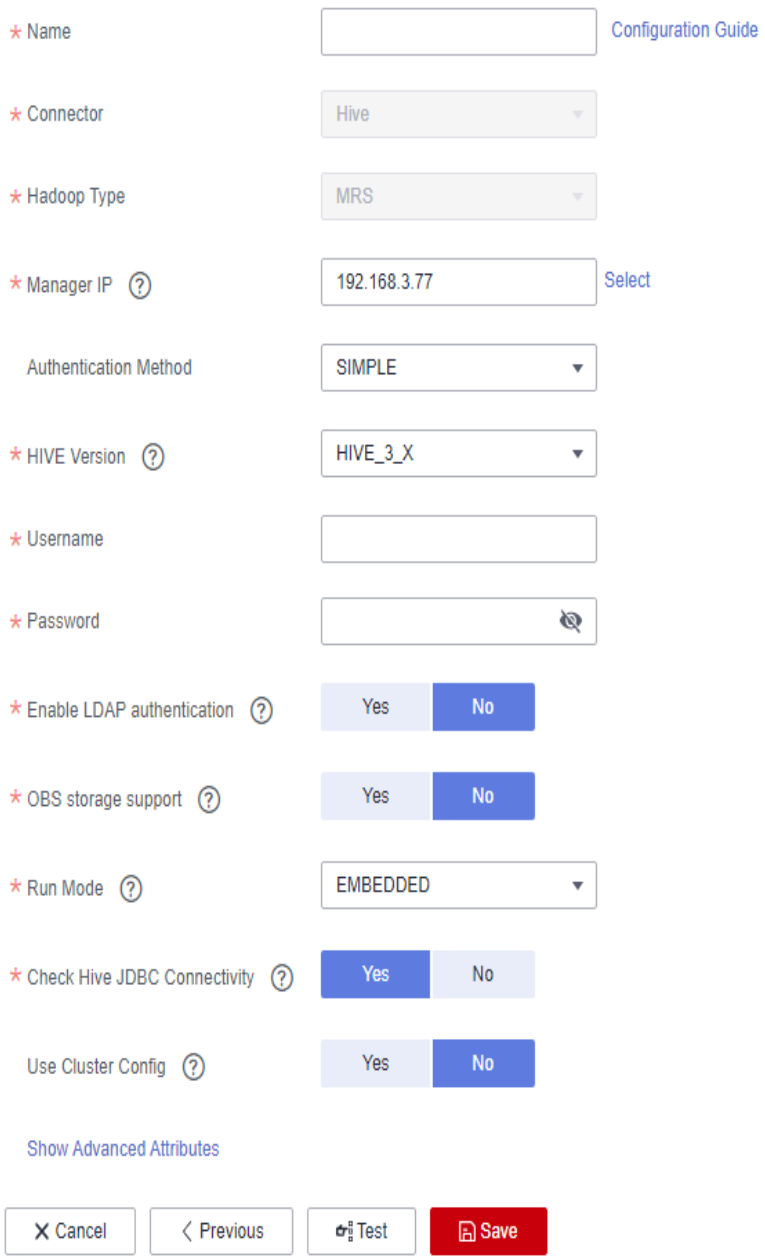
Creating an MRS Hive Link

Step 1 On the **Cluster Management** page, locate a cluster and click **Job Management** in the **Operation** column. On the displayed page, click the **Links** tab and then **Create Link**.

Figure 8-1 Selecting a connector type

Step 2 Select **MRS Hive** and click **Next** to configure parameters for the MRS Hive link.

Figure 8-2 Creating an MRS Hive link



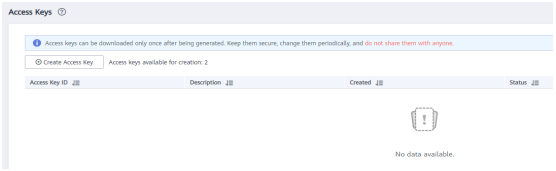
Step 3 Click **Show Advanced Attributes** to view more optional parameters. Retain their default values. The following table lists the mandatory parameters.

Table 8-1 MRS Hive link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	hivelink

Parameter	Description	Example Value
Manager IP	Floating IP address of MRS Manager. Click **Select** next to the **Manager IP** text box to select an MRS cluster. CDM automatically fills in the authentication information.	127.0.0.1
Authentication Method	Authentication method used for accessing MRS • **SIMPLE**: Select this for non-security mode. • **KERBEROS**: Select this for security mode.	SIMPLE
HIVE Version	Set this to the Hive version on the server.	HIVE_3_X
Username	If **Authentication Method** is set to **KERBEROS**, you must provide the username and password used for logging in to MRS Manager. If you need to create a snapshot when exporting a directory from HDFS, the user configured here must have the administrator permission on HDFS. To create a data connection for an MRS security cluster, do not use user **admin**. The **admin** user is the default management page user and cannot be used as the authentication user of the security cluster. You can create an MRS user and set **Username** and **Password** to the username and password of the created MRS user when creating an MRS data connection. **NOTE** • If the CDM cluster version is 2.9.0 or later and the MRS cluster version is 3.1.0 or later, the created user must have the permissions of the **Manager_viewer** role to create links on CDM. To perform operations on databases, tables, and columns of an MRS component, you also need to add the database, table, and column permissions of the MRS component to the user by following the instructions in the MRS documentation. • If the CDM cluster version is earlier than 2.9.0 or the MRS cluster version is earlier than 3.1.0, the created user must have the permissions of **Manager_administrator** or **System_administrator** to create links on CDM. • A user with only the **Manager_tenant** or **Manager_auditor** permission cannot create connections.	cdm
Password	Password used for logging in to MRS Manager	-

Parameter	Description	Example Value
Enable ldap	This parameter is available when **Proxy connection** is selected for **Connection Type**. If LDAP authentication is enabled for an external LDAP server connected to MRS Hive, the LDAP username and password are required for authenticating the connection to MRS Hive. In this case, this option must be enabled. Otherwise, the connection will fail.	No
ldapUsername	This parameter is mandatory when **Enable ldap** is enabled. Enter the username configured when LDAP authentication was enabled for MRS Hive.	-
ldapPassword	This parameter is mandatory when **Enable ldap** is enabled. Enter the password configured when LDAP authentication was enabled for MRS Hive.	-
OBS storage support	The server must support OBS storage. When creating a Hive table, you can store the table in OBS.	No

Parameter	Description	Example Value
AK	This parameter is mandatory when **OBS storage support** is enabled. The account corresponding to the AK/SK pair must have the OBS Buckets Viewer permission. Otherwise, OBS cannot be accessed and the "403 AccessDenied" error is reported. You need to create an access key for the current account and obtain an AK/SK pair. 1. Log in to the management console, move the cursor to the username in the upper right corner, and select **My Credentials** from the drop-down list. 2. On the **My Credentials** page, choose **Access Keys**, and click **Create Access Key**. See **Figure 8-3**. **Figure 8-3** Clicking Create Access Key  3. Click **OK** and save the access key file as prompted. The access key file will be saved to your browser's configured download location. Open the **credentials.csv** file to view **Access Key Id** and **Secret Access Key**. **NOTE** – Only two access keys can be added for each user. – To ensure access key security, the access key is automatically downloaded only when it is generated for the first time and cannot be obtained from the management console later. Keep them properly.	-
SK		-

Parameter	Description	Example Value
Run Mode	This parameter is used only when the Hive version is **HIVE_3_X**. Possible values are: • **EMBEDDED**: The link instance runs with CDM. This mode delivers better performance. • **Standalone**: The link instance runs in an independent process. If CDM needs to connect to multiple Hadoop data sources (MRS, Hadoop, or CloudTable) with both Kerberos and Simple authentication modes, **Standalone** prevails. **NOTE** The **STANDALONE** mode is used to solve the version conflict problem. If the connector versions of the source and destination ends of the same link are different, a JAR file conflict occurs. In this case, you need to place the source or destination end in the STANDALONE process to prevent the migration failure caused by the conflict.	EMBEDDED
Check Hive JDBC Connectivity	Whether to check the Hive JDBC connectivity	No
Use Cluster Config	You can use the cluster configuration to simplify parameter settings for the Hadoop connection.	No
Cluster Config Name	This parameter is valid only when **Use Cluster Config** is set to **Yes**. Select a cluster configuration that has been created. For details about how to configure a cluster, see **Managing Cluster Configurations**.	hive_01

NOTE

Click **Show Advanced Attributes**, and then click **Add** to add configuration attributes of other clients. The name and value of each attribute must be configured. You can click **Delete** to delete no longer used attributes.

Step 4 Click **Save** to return to the **Links**page.

----End

8.2 Creating a MySQL Link

MySQL links are applicable to third-party cloud MySQL services and MySQL created in a local data center or ECS. This tutorial describes how to create a MySQL link.

Prerequisites

- You have obtained the IP address, port, database name, username, and password for connecting to the MySQL database. In addition, the user must have the read and write permissions on the MySQL database.
- The on-premises MySQL database can be accessed through the public network. If the MySQL database is deployed on an on-premises data center or a third-party cloud, ensure that an IP address that can be accessed from the public network has been configured for the MySQL database, or the VPN or Direct Connect between the on-premises data center and the cloud service platform has been established.
- You have created a CDM cluster.

Creating a MySQL Link

Step 1 Access the CDM console, choose **Cluster Management** in the navigation pane, locate the target cluster, and choose **Job Management** > **Link Management** > **Driver Management**. The **Driver Management** page is displayed.

Step 2 On the **Driver Management** page, click the document link in the **Recommended Version** column of the MySQL driver and obtain the driver file as instructed.

Step 3 On the **Driver Management** page, upload the MySQL driver using either of the following methods:

Click **Upload** in the **Operation** column and select a local driver.

Alternatively, click **Copy from SFTP** in the **Operation** column and configure the **SFTP Link** name and **Driver File Path**.

Step 4 On the **Cluster Management** page, click **Job Management** of the cluster and choose **Links** > **Create Link** to enter the page for selecting the connector.

Figure 8-4 Selecting a connector type

Step 5 Select **MySQL** and click **Next** to configure parameters for the MySQL link.

Table 8-2 MySQL link parameters

Parameter	Description	Example Value
Name	Enter a unique link name.	mysqllink
Database Server	IP address or domain name of the MySQL database	192.168.1.110
Port	MySQL database port	3306
Database Name	Name of the MySQL database	sqoop
Username	User who has the read, write, and delete permissions on the MySQL database	admin
Password	Password of the user	-
Use Local API	Whether to use the local API of the database for acceleration. (The system attempts to enable the **local_infile** system variable of the MySQL database.)	Yes
Use Agent	Whether to extract data from the data source through an agent	Yes
local_infile Character Set	When using local_infile to import data to MySQL, you can configure the encoding format.	utf8
Driver Version	A driver version that adapts to MySQL	-
Agent	Click **Select** and select the created agent.	-
Fetch Size	Number of rows obtained by each request	1000
Commit Size	Obtaining data from the source through the agent	1000
Link Attributes	Custom attributes of the link	useCompression=true
Reference Sign	Delimiter used to separate referenced table names or column names This parameter is left blank by default.	'
Batch Size	Number of rows written each time. It should be less than **Commit Size**. When the number of rows written reaches the value of **Commit Size**, the rows will be committed to the database.	100

Step 6 Click **Save** to return to the **Links** page.

NOTE

If an error occurs during the saving, the security settings of the MySQL database are incorrect. In this case, you need to enable the EIP of the CDM cluster to access the MySQL database.

----End

8.3 Migrating Data from MySQL to MRS Hive

MRS provides enterprise-level big data clusters on the cloud. It contains HDFS, Hive, and Spark components and is applicable to massive data analysis of enterprises.

Hive supports SQL to help users perform extraction, transformation, and loading (ETL) operations on large-scale data sets. Query on large-scale data sets takes a long time. In many scenarios, you can create Hive partitions to reduce the total amount of data to be scanned each time. This significantly improves query performance.

Hive partitions are implemented by using the HDFS subdirectory function. Each subdirectory contains the column names and values of each partition. If there are multiple partitions, many HDFS subdirectories exist. It is not easy to load external data to each partition of the Hive table without relying on tools. With CDM, you can easily load data of the external data sources (relational databases, object storage services, and file system services) to Hive partition tables.

This section describes how to migrate data from the MySQL database to the MRS Hive partition table.

Scenario

Suppose that there is a **trip_data** table in the MySQL database. The table stores cycling records such as the start time, end time, start sites, end sites, and rider IDs. For details about the fields in the **trip_data** table, see **Figure 8-5**.

Figure 8-5 MySQL table fields

Column Name	#	Data Type
TripID	1	int(11)
Duration	2	int(11)
StartDate	3	timestamp
StartStation	4	varchar(64)
StartTerminal	5	int(11)
EndDate	6	timestamp
EndStation	7	varchar(64)
EndTerminal	8	int(11)
Bike	9	int(11)
SubscriberType	10	varchar(32)
ZipCodev	11	varchar(10)

The following describes how to use CDM to import the **trip_data** table in the MySQL database to the MRS Hive partition table. The procedure is as follows:

1. **Creating a Hive Partition Table on MRS Hive**
2. **Creating a CDM Cluster and Binding an EIP to the Cluster**
3. **Creating a MySQL Link**
4. **Creating a Hive Link**
5. **Creating a Migration Job**

Prerequisites

- MRS is available.
- You have obtained the IP address, port, database name, username, and password for connecting to the MySQL database. In addition, the user must have the read and write permissions on the MySQL database.
- You have uploaded the MySQL database driver on the **Job Management** > **Links** > **Driver Management** page.

Creating a Hive Partition Table on MRS Hive

On MRS Hive, run the following SQL statement to create a Hive partition table named **trip_data** with three new fields **y**, **ym**, and **ymd** used as partition fields. The SQL statement is as follows:

```
create table trip_data(TripID int,Duration int,StartDate timestamp,StartStation varchar(64),StartTerminal int,EndDate timestamp,EndStation varchar(64),EndTerminal int,Bike int,SubscriberType varchar(32),ZipCodev varchar(10))partitioned by (y int,ym int,ymd int);
```

NOTE

The **trip_data** partition table has three partition fields: year, year and month, and year, month, and date of the start time of a ride. For example, if the start time of a ride is **2018/5/11 9:40**, the record is saved in the **trip_data/2018/201805/20180511** partition. When the records in the **trip_data** table are summarized, only part of the data needs to be scanned, improving the performance.

Creating a CDM Cluster and Binding an EIP to the Cluster

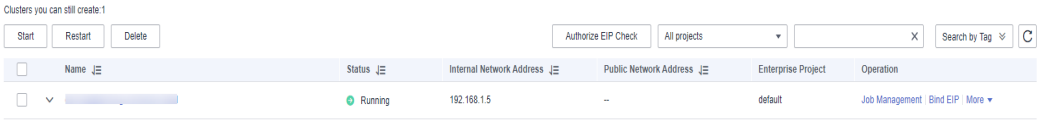
Step 1 If CDM is used an independent service, create a CDM cluster by following the instructions in **Creating a CDM Cluster**. If CDM is used as a module of DataArts Studio, create a CDM cluster by following the instructions in **Creating a CDM Cluster**.

The key configurations are as follows:

- The flavor of the CDM cluster is selected based on the amount of data to be migrated. Generally, cdm.medium meets the requirements for most migration scenarios.
- The CDM and MRS clusters must be in the same VPC, subnet, and security group.

Step 2 After the CDM cluster is created, on the **Cluster Management** page, click **Bind EIP** in the **Operation** column to bind an EIP to the cluster. The CDM cluster uses the EIP to access MySQL.

Figure 8-6 Cluster list



NOTE

If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

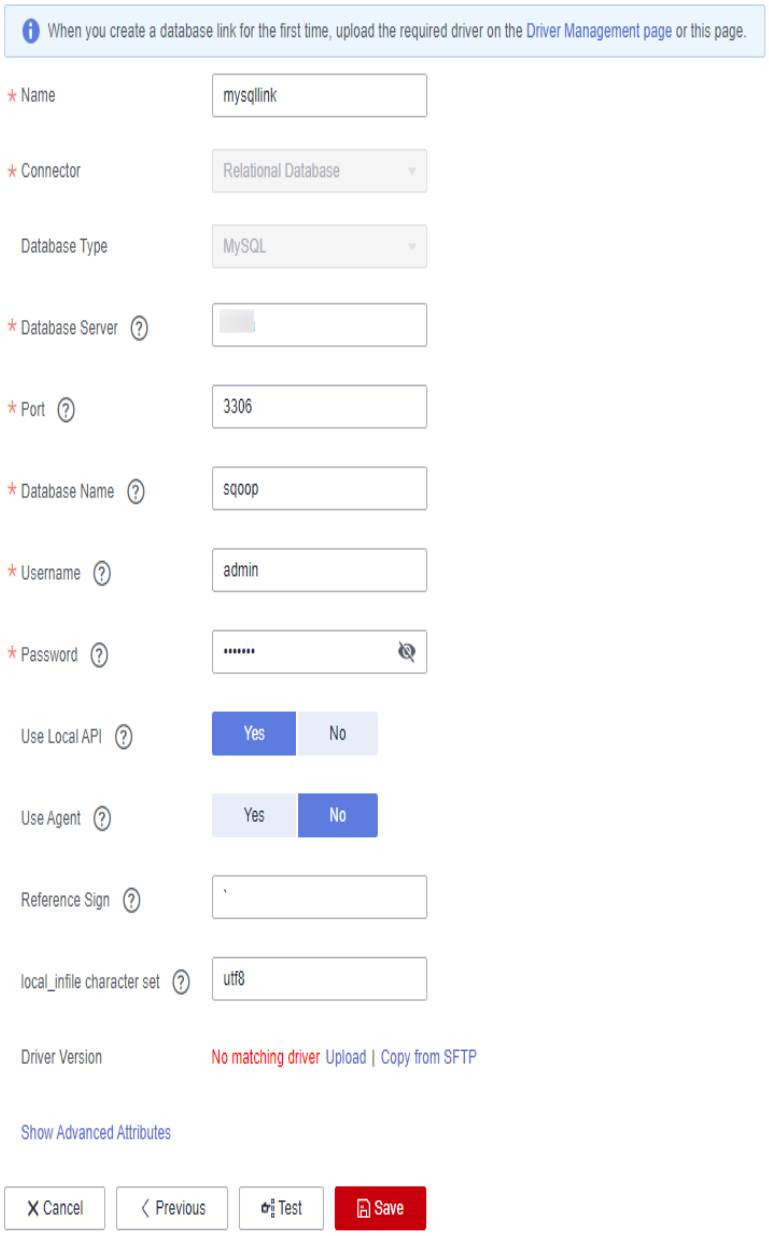
Creating a MySQL Link

Step 1 On the **Cluster Management** page, locate a cluster and click **Job Management** in the **Operation** column. On the displayed page, click the **Links** tab and then **Create Link**.

Figure 8-7 Selecting a connector

Step 2 Select **RDS for MySQL** and click **Next** to set the link parameters.

Figure 8-8 Creating a MySQL link



Click **Show Advanced Attributes** to view more optional parameters. For details, see **Link to an RDS for MySQL/MySQL Database**. Retain the default values of the optional parameters and configure the mandatory parameters according to **Table 8-3**.

Table 8-3 MySQL link parameters

Parameter	Description	Example Value
Name	Unique link name	mysqllink
Database Server	IP address or domain name of the MySQL database server	-

Parameter	Description	Example Value
Port	MySQL database port	3306
Database Name	Name of the MySQL database	sqoop
Username	User who has the read, write, and delete permissions on the MySQL database	admin
Password	Password of the user	-
Use Local API	Whether to use the local API of the database for acceleration. (The system attempts to enable the **local_infile** system variable of the MySQL database.)	Yes
Use Agent	Whether to extract data from the data source through an agent	No
local_infile Character Set	When using local_infile to import data to MySQL, you can configure the encoding format.	utf8
Driver Version	Before connecting CDM to a relational database, you need to upload the JDK 8 .jar driver of the relational database. Download the MySQL driver 5.1.48 from **https://downloads.mysql.com/archives/c-j/**, obtain **mysql-connector-java-5.1.48.jar**, and upload it.	-

Step 3 Click **Save**. The **Link Management** page is displayed.

NOTE

If an error occurs during the saving, the security settings of the MySQL database are incorrect. In this case, you need to enable the EIP of the CDM cluster to access the MySQL database.

----End

Creating a Hive Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 8-9 Selecting a connector type

Step 2 Select **MRS Hive** and click **Next** to configure parameters for the MRS Hive link.

Figure 8-10 Creating an MRS Hive link

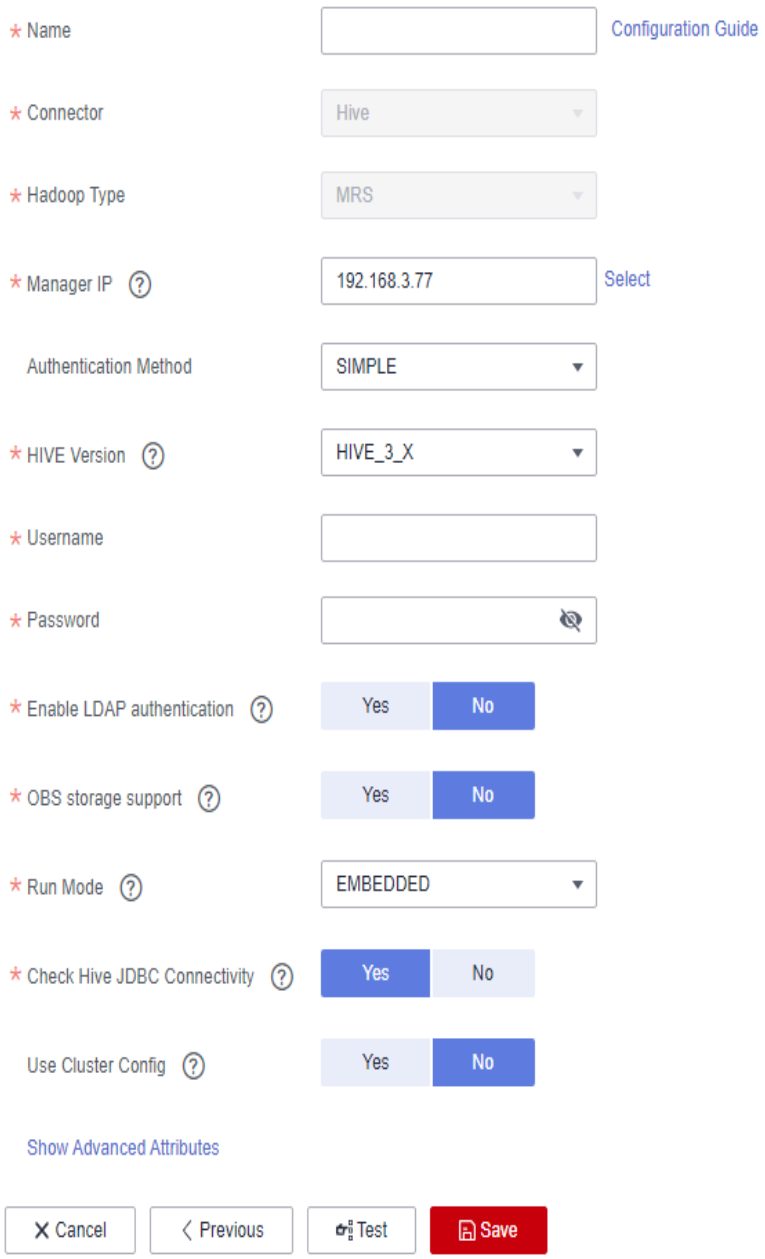


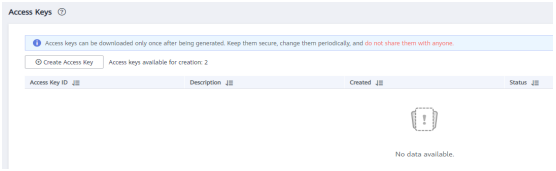
Table 8-4 describes the parameters. You can configure the parameters according to the actual situation.

Table 8-4 MRS Hive link parameters

Parameter	Description	Example Value
Name	Link name, which should be defined based on the data source type, so it is easier to remember what the link is for	hivelink

Parameter	Description	Example Value
Manager IP	Floating IP address of MRS Manager. Click **Select** next to the **Manager IP** text box to select an MRS cluster. CDM automatically fills in the authentication information.	127.0.0.1
Authentication Method	Authentication method used for accessing MRS • **SIMPLE**: Select this for non-security mode. • **KERBEROS**: Select this for security mode.	SIMPLE
HIVE Version	Set this to the Hive version on the server.	HIVE_3_X
Username	If **Authentication Method** is set to **KERBEROS**, you must provide the username and password used for logging in to MRS Manager. If you need to create a snapshot when exporting a directory from HDFS, the user configured here must have the administrator permission on HDFS. To create a data connection for an MRS security cluster, do not use user **admin**. The **admin** user is the default management page user and cannot be used as the authentication user of the security cluster. You can create an MRS user and set **Username** and **Password** to the username and password of the created MRS user when creating an MRS data connection. **NOTE** • If the CDM cluster version is 2.9.0 or later and the MRS cluster version is 3.1.0 or later, the created user must have the permissions of the **Manager_viewer** role to create links on CDM. To perform operations on databases, tables, and columns of an MRS component, you also need to add the database, table, and column permissions of the MRS component to the user by following the instructions in the MRS documentation. • If the CDM cluster version is earlier than 2.9.0 or the MRS cluster version is earlier than 3.1.0, the created user must have the permissions of **Manager_administrator** or **System_administrator** to create links on CDM. • A user with only the **Manager_tenant** or **Manager_auditor** permission cannot create connections.	cdm
Password	Password used for logging in to MRS Manager	-

Parameter	Description	Example Value
Enable ldap	This parameter is available when **Proxy connection** is selected for **Connection Type**. If LDAP authentication is enabled for an external LDAP server connected to MRS Hive, the LDAP username and password are required for authenticating the connection to MRS Hive. In this case, this option must be enabled. Otherwise, the connection will fail.	No
ldapUsername	This parameter is mandatory when **Enable ldap** is enabled. Enter the username configured when LDAP authentication was enabled for MRS Hive.	-
ldapPassword	This parameter is mandatory when **Enable ldap** is enabled. Enter the password configured when LDAP authentication was enabled for MRS Hive.	-
OBS storage support	The server must support OBS storage. When creating a Hive table, you can store the table in OBS.	No

Parameter	Description	Example Value
AK	This parameter is mandatory when **OBS storage support** is enabled. The account corresponding to the AK/SK pair must have the OBS Buckets Viewer permission. Otherwise, OBS cannot be accessed and the "403 AccessDenied" error is reported. You need to create an access key for the current account and obtain an AK/SK pair. 1. Log in to the management console, move the cursor to the username in the upper right corner, and select **My Credentials** from the drop-down list. 2. On the **My Credentials** page, choose **Access Keys**, and click **Create Access Key**. See **Figure 8-11**. **Figure 8-11** Clicking Create Access Key  3. Click **OK** and save the access key file as prompted. The access key file will be saved to your browser's configured download location. Open the **credentials.csv** file to view **Access Key Id** and **Secret Access Key**. **NOTE** – Only two access keys can be added for each user. – To ensure access key security, the access key is automatically downloaded only when it is generated for the first time and cannot be obtained from the management console later. Keep them properly.	-
SK		-

Parameter	Description	Example Value
Run Mode	This parameter is used only when the Hive version is **HIVE_3_X**. Possible values are: • **EMBEDDED**: The link instance runs with CDM. This mode delivers better performance. • **Standalone**: The link instance runs in an independent process. If CDM needs to connect to multiple Hadoop data sources (MRS, Hadoop, or CloudTable) with both Kerberos and Simple authentication modes, **Standalone** prevails. **NOTE** The **STANDALONE** mode is used to solve the version conflict problem. If the connector versions of the source and destination ends of the same link are different, a JAR file conflict occurs. In this case, you need to place the source or destination end in the STANDALONE process to prevent the migration failure caused by the conflict.	EMBEDDED
Check Hive JDBC Connectivity	Whether to check the Hive JDBC connectivity	No
Use Cluster Config	You can use the cluster configuration to simplify parameter settings for the Hadoop connection.	No
Cluster Config Name	This parameter is valid only when **Use Cluster Config** is set to **Yes**. Select a cluster configuration that has been created. For details about how to configure a cluster, see **Managing Cluster Configurations**.	hive_01

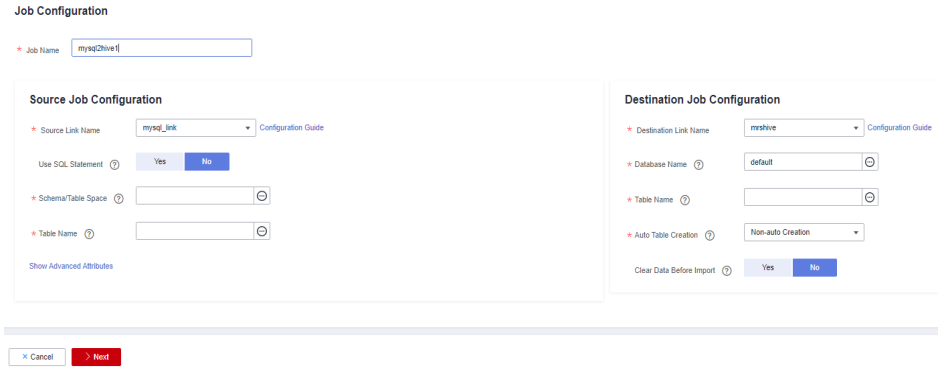
Step 3 Click **Save**. The **Link Management** page is displayed.

----End

Creating a Migration Job

Step 1 Click the **Table/File Migration** tab and then **Create Job**.

Figure 8-12 Creating a job for migrating data from MySQL to Hive



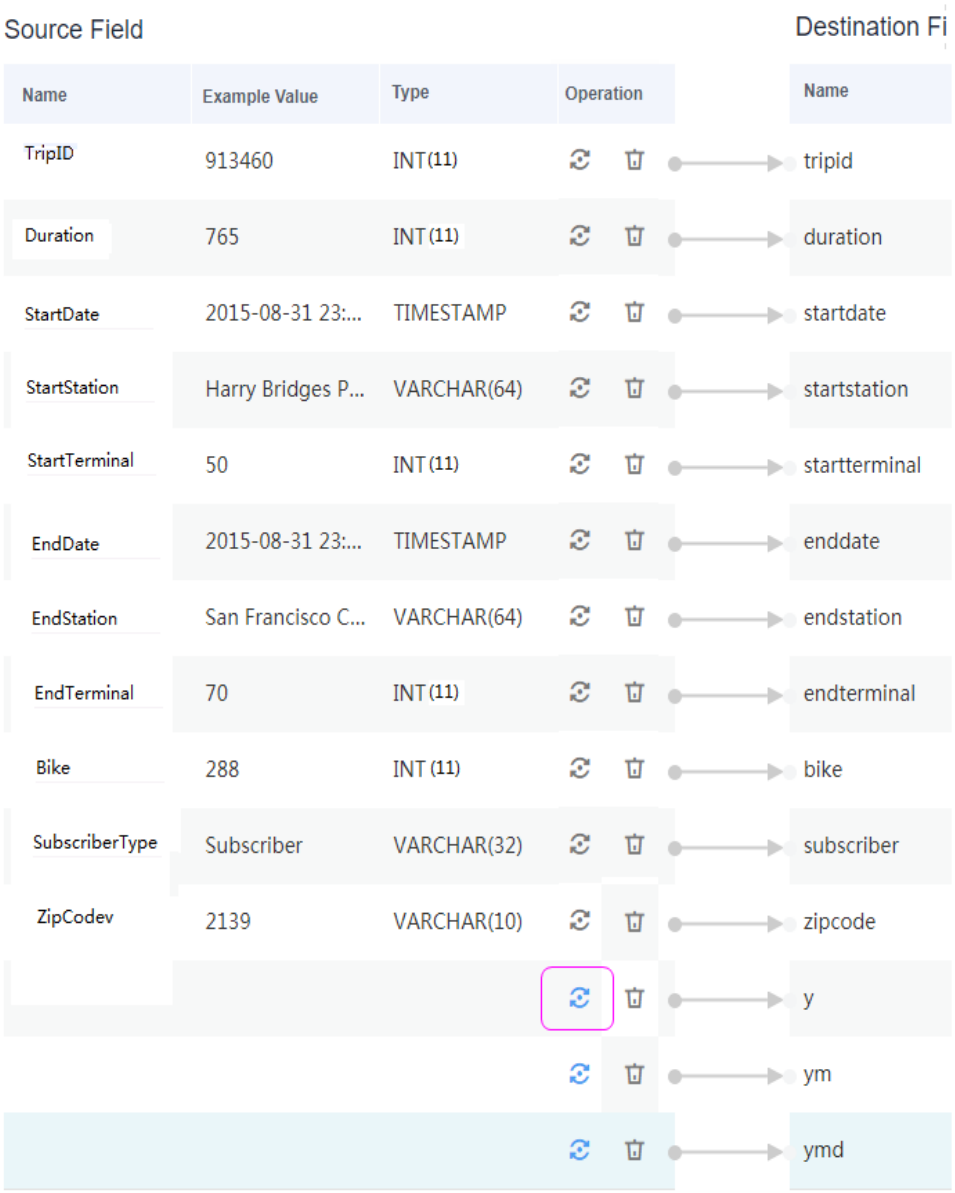
NOTE

Set **Clear Data Before Import** to **Yes**, so that the data in the Hive table will be cleared before data import.

Step 2 After the parameters are configured, click **Next**. The **Map Field** tab page is displayed. See **Figure 8-13**.

Map the fields of the MySQL table and Hive table. The Hive table has three more fields **y**, **ym**, and **ymd** than the MySQL table, which are the Hive partition fields. Because the fields of the source table cannot be directly mapped to the destination table, you need to configure an expression to extract data from the **StartDate** field in the source table.

Figure 8-13 Hive field mapping



Step 3 Click to display the **Converter List** dialog box, and then choose **Create Converter** > **Expression conversion**. See **Figure 8-14**.

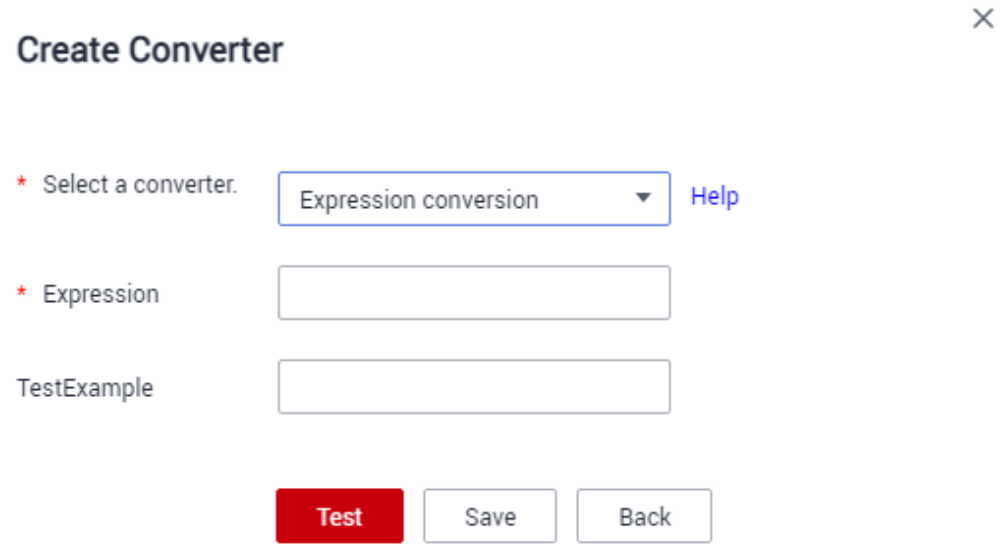
The expressions for the **y**, **ym**, and **ymd** fields are as follows:

DateUtils.format(DateUtils.parseDate(row[2],"yyyy-MM-dd HH:mm:ss.SSS"),"yyyy")

DateUtils.format(DateUtils.parseDate(row[2],"yyyy-MM-dd HH:mm:ss.SSS"),"yyyyMM")

DateUtils.format(DateUtils.parseDate(row[2],"yyyy-MM-dd HH:mm:ss.SSS"),"yyyyMMdd")

Figure 8-14 Configuring the expression



NOTE

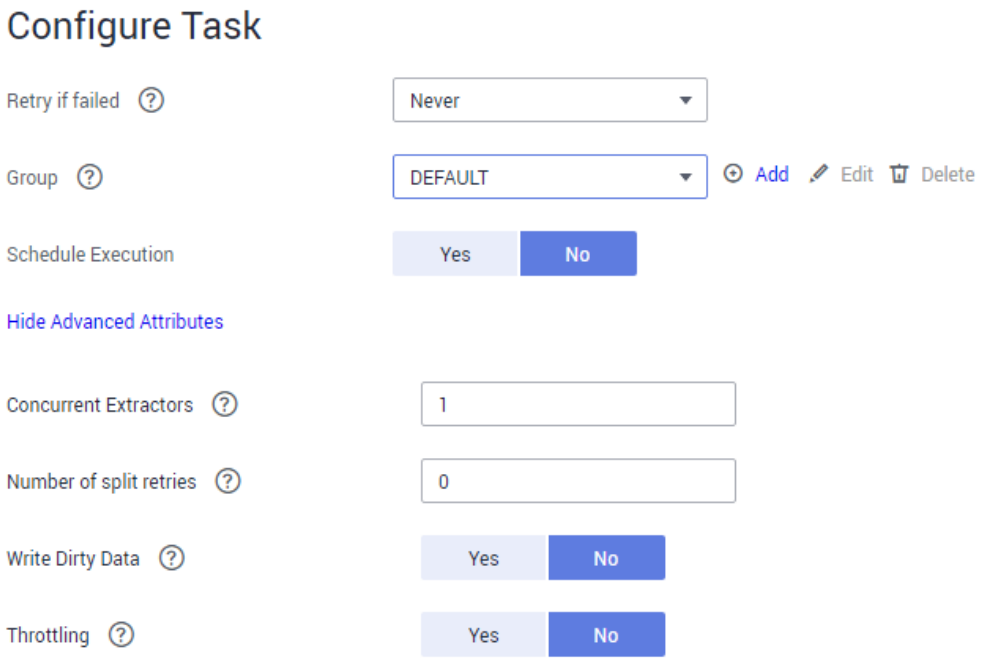
The expressions in CDM support field conversion of common character strings, dates, and values. For details, see **Converting Fields**.

Step 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 If Failed**: Determine whether to automatically retry the job if it fails. 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.
- **Schedule Execution**: Determine whether to automatically execute the job at a scheduled time. Retain the default value **No** in this example.
- **Concurrent Extractors**: Enter the number of concurrent extractors. An appropriate value improves migration efficiency. For details, see **Performance Tuning**. 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 for future viewing. Before writing dirty data, create an OBS link on the CDM console. Retain the default value **No** so that dirty data is not recorded.

Figure 8-15 Configuring the task



Step 5 Click **Save and Run**. The **Job Management** page is displayed, on which you can view the job execution progress and result.

Step 6 After the job is successfully executed, in the **Operation** column of the job, click **Historical Record** to view the job's historical execution records and read/write statistics.

On the **Historical Record** page, click **Log** to view the job logs.

----End

8.4 Migrating Data from MySQL to OBS

Scenario

CDM supports table-to-OBS data migration. This section describes how to migrate tables from a MySQL database to OBS. The process is as follows:

1. **Creating a CDM Cluster and Binding an EIP to the Cluster**
2. **Creating a MySQL Link**
3. **Creating an OBS Link**
4. **Creating a Migration Job**

Prerequisites

- You have obtained the domain name, port number, AK, and SK for accessing OBS.
- You have obtained the IP address, port, database name, username, and password for connecting to the MySQL database. In addition, the user must have the read and write permissions on the MySQL database.

- You have uploaded the MySQL database driver on the **Job Management** > **Links** > **Driver Management** page.

Creating a CDM Cluster and Binding an EIP to the Cluster

Step 1 If CDM is used an independent service, create a CDM cluster by following the instructions in **Creating a CDM Cluster**. If CDM is used as a module of DataArts Studio, create a CDM cluster by following the instructions in **Creating a CDM Cluster**.

The key configurations are as follows:

The flavor of the CDM cluster is selected based on the amount of data to be migrated. Generally, cdm.medium meets the requirements for most migration scenarios.

Step 2 After the CDM cluster is created, on the **Cluster Management** page, click **Bind EIP** in the **Operation** column to bind an EIP to the cluster. The CDM cluster uses the EIP to access MySQL.

NOTE

If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

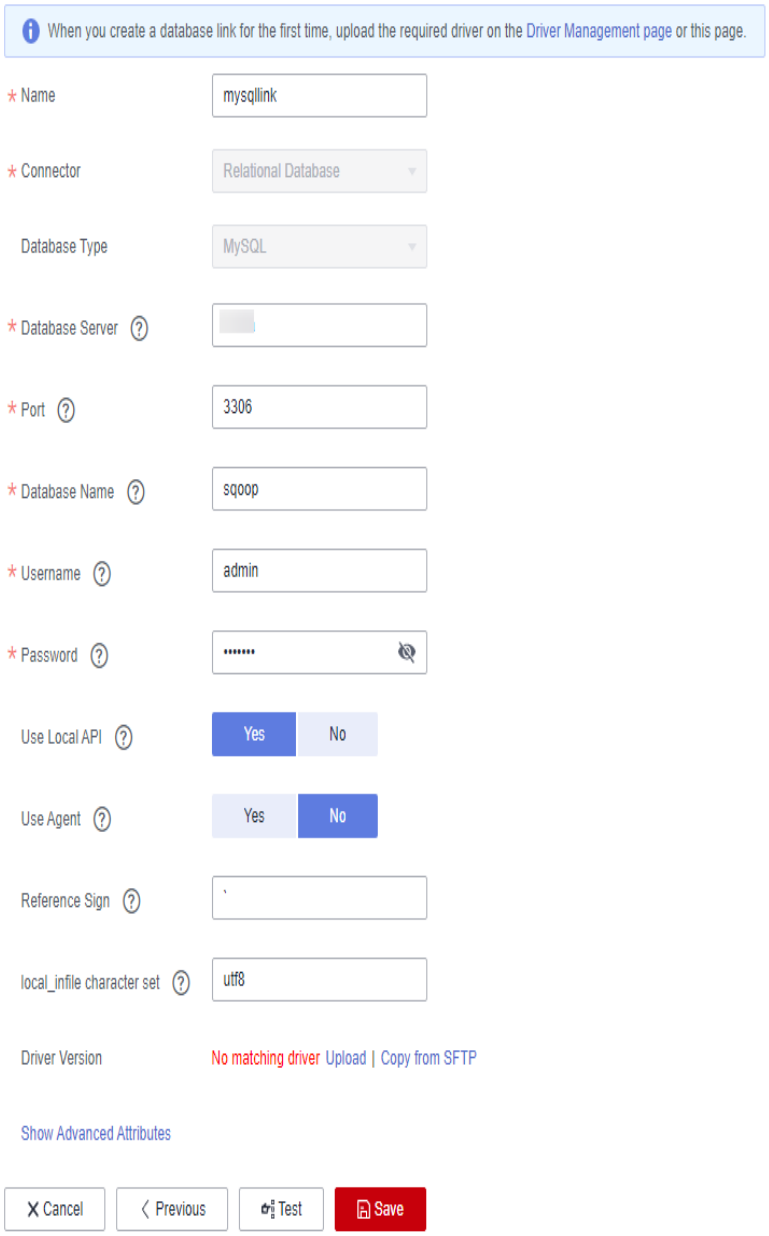
Creating a MySQL Link

Step 1 On the **Cluster Management** page, locate a cluster and click **Job Management** in the **Operation** column. On the displayed page, click the **Links** tab and then **Create Link**.

Figure 8-16 Selecting a connector

Step 2 Select **RDS for MySQL** and click **Next** to set the link parameters.

Figure 8-17 Creating a MySQL link



Click **Show Advanced Attributes** to view more optional parameters. For details, see **Link to an RDS for MySQL/MySQL Database**. Retain the default values of the optional parameters and configure the mandatory parameters according to **Table 8-5**.

Table 8-5 MySQL link parameters

Parameter	Description	Example Value
Name	Unique link name	mysqllink

Parameter	Description	Example Value
Database Server	IP address or domain name of the MySQL database server	-
Port	MySQL database port	3306
Database Name	Name of the MySQL database	sqoop
Username	User who has the read, write, and delete permissions on the MySQL database	admin
Password	Password of the user	-
Use Local API	Whether to use the local API of the database for acceleration. (The system attempts to enable the **local_infile** system variable of the MySQL database.)	Yes
Use Agent	Whether to extract data from the data source through an agent	No
local_infile Character Set	When using local_infile to import data to MySQL, you can configure the encoding format.	utf8
Driver Version	Before connecting CDM to a relational database, you need to upload the JDK 8 .jar driver of the relational database. Download the MySQL driver 5.1.48 from **https://downloads.mysql.com/archives/c-j/**, obtain **mysql-connector-java-5.1.48.jar**, and upload it.	-

Step 3 Click **Save**. The **Link Management** page is displayed.

NOTE

If an error occurs during the saving, the security settings of the MySQL database are incorrect. In this case, you need to enable the EIP of the CDM cluster to access the MySQL database.

----End

Creating an OBS Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 8-18 Selecting a connector type

Step 2 Select **Object Storage Service (OBS)** and click **Next** to configure parameters for the OBS link.

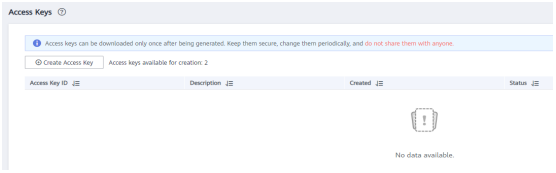
- **Name**: Enter a custom link name, for example, **obslink**.
- **OBS Server** and **Port**: Enter the actual OBS address information.
- **AK** and **SK**: Enter the AK and SK used for logging in to OBS.

  To obtain an access key, perform the following steps:

  a. Log in to the management console, move the cursor to the username in the upper right corner, and select **My Credentials** from the drop-down list.

  b. On the **My Credentials** page, choose **Access Keys**, and click **Create Access Key**. See **Figure 8-19**.

  Figure 8-19 Clicking Create Access Key

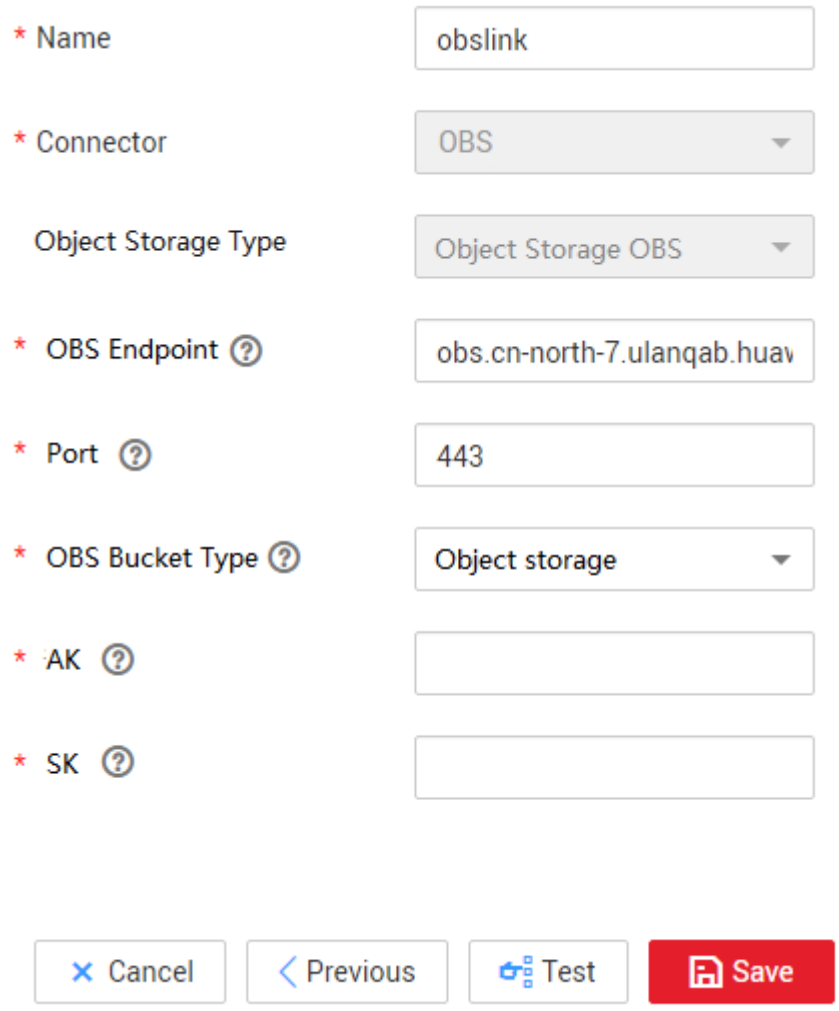
  

  c. Click **OK** and save the access key file as prompted. The access key file will be saved to your browser's configured download location. Open the **credentials.csv** file to view **Access Key Id** and **Secret Access Key**.

  NOTE

  - Only two access keys can be added for each user.
  - To ensure access key security, the access key is automatically downloaded only when it is generated for the first time and cannot be obtained from the management console later. Keep them properly.

Figure 8-20 Creating an OBS link



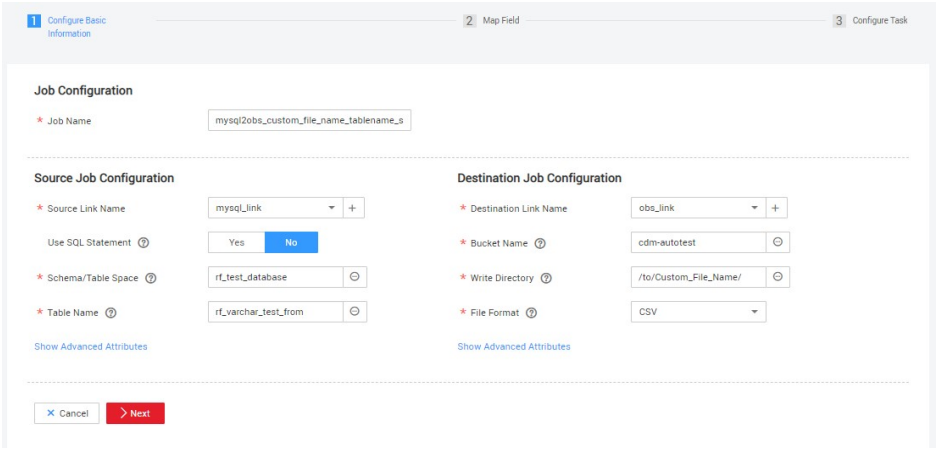
Step 3 Click **Save**. The **Link Management** page is displayed.

----End

Creating a Migration Job

Step 1 Choose **Table/File Migration** > **Create Job** to create a job for exporting data from the MySQL database to OBS.

Figure 8-21 Creating a job for migrating data from MySQL to OBS

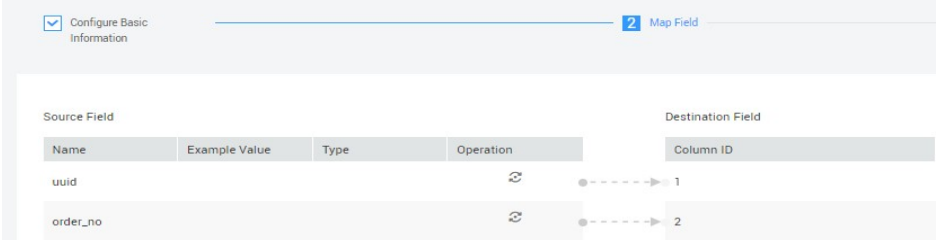


- **Job Name**: Enter a unique name.
- **Source Job Configuration**
  - **Source Link Name**: Select the **mysqllink** created in **Creating a MySQL Link**.
  - **Use SQL Statement**: Select **No**.
  - **Schema/Tablespace**: name of the schema or tablespace from which data is to be extracted
  - **Table Name**: name of the table from which data is to be extracted
  - Retain the default values of other optional parameters.
- **Destination Job Configuration**
  - **Destination Link Name**: Select the **obslink** created in **Creating an OBS Link**.
  - **Bucket Name**: Select the bucket from which the data will be migrated.
  - **Write Directory**: Enter the directory to which data is to be written on the OBS server.
  - **File Format**: Select **CSV**.
  - Retain the default values of the optional parameters in **Show Advanced Attributes**.

Step 2 Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields, as shown in **Figure 8-22**.

- If the field mapping is incorrect, you can drag the fields to adjust the mapping.
- The expressions in CDM support field conversion of common character strings, dates, and values. For details, see **Converting Fields**.

Figure 8-22 Table-to-file field mapping



Step 3 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.
- **Schedule Execution**: Enable it if you need to configure scheduled jobs. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. CDM supports concurrent extraction of MySQL data. If indexes are configured for the source table, you can increase the number of concurrent extractors to accelerate the migration.
- **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 for future viewing. Before writing dirty data, create an OBS link. For file-to-table data migration, you are advised to write dirty data.
- **Delete Job After Completion**: Retain the default value **Do not delete**. You can also set this parameter to **Delete** to prevent an accumulation of too many migration jobs.

Step 4 Click **Save and Run**. The **Job Management** page is displayed, on which you can view the job execution progress and result.

Step 5 After the job is successfully executed, in the **Operation** column of the job, click **Historical Record** to view the job's historical execution records and read/write statistics.

On the **Historical Record** page, click **Log** to view the job logs.

----End

8.5 Migrating Data from MySQL to DWS

Scenario

CDM supports table-to-table data migration. This section describes how to migrate data from MySQL to DWS. The process is as follows:

1. **Creating a CDM Cluster and Binding an EIP to the Cluster**
2. **Creating a MySQL Link**
3. **Creating a DWS Link**
4. **Creating a Migration Job**

Prerequisites

- You have obtained the IP address, port number, database name, username, and password for connecting to DWS. In addition, you must have the read, write, and delete permissions on the DWS database.

- You have obtained the IP address, port, database name, username, and password for connecting to the MySQL database. In addition, the user must have the read and write permissions on the MySQL database.
- You have uploaded the MySQL database driver on the **Job Management** > **Links** > **Driver Management** page.

Creating a CDM Cluster and Binding an EIP to the Cluster

Step 1 If CDM is used an independent service, create a CDM cluster by following the instructions in **Creating a CDM Cluster**. If CDM is used as a module of DataArts Studio, create a CDM cluster by following the instructions in **Creating a CDM Cluster**.

The key configurations are as follows:

- The flavor of the CDM cluster is selected based on the amount of data to be migrated. Generally, cdm.medium meets the requirements for most migration scenarios.
- The VPC, subnet, and security group of the CDM cluster must be the same as those of the DWS cluster.

Step 2 After the CDM cluster is created, on the **Cluster Management** page, click **Bind EIP** in the **Operation** column to bind an EIP to the cluster. The CDM cluster uses the EIP to access MySQL.

NOTE

If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

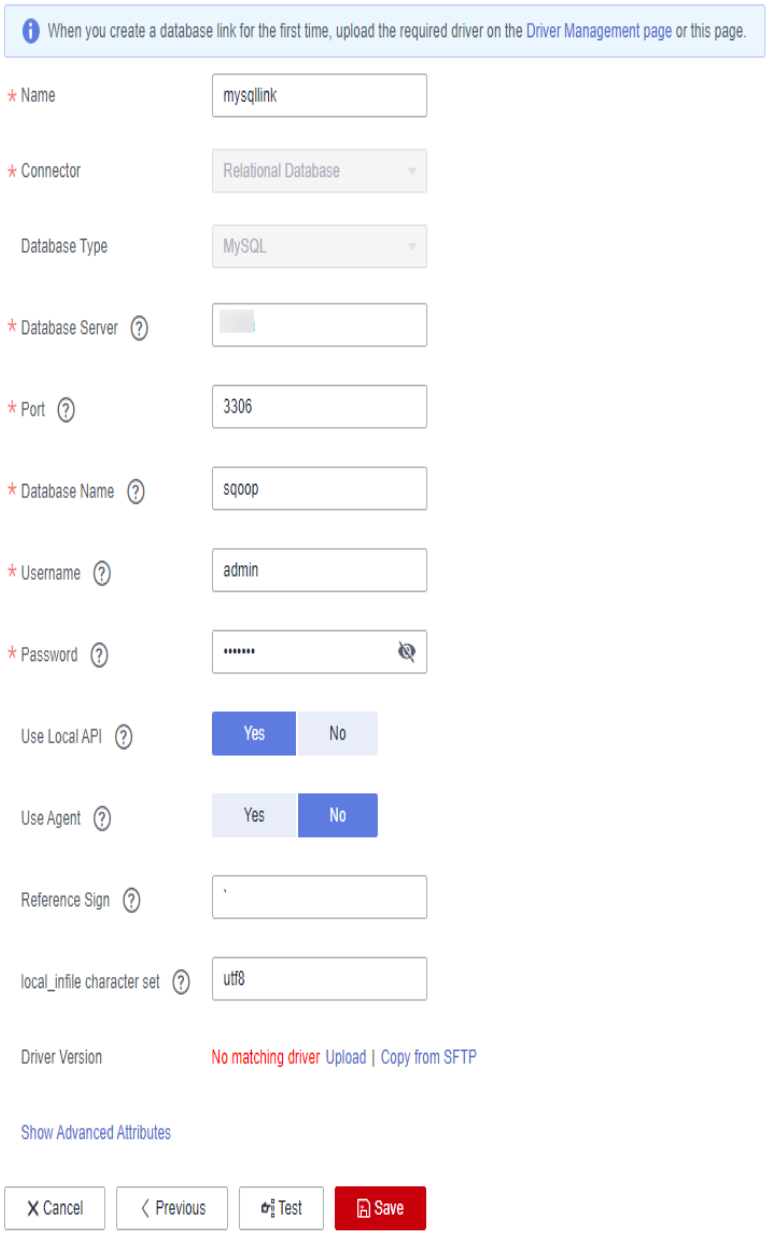
Creating a MySQL Link

Step 1 On the **Cluster Management** page, locate a cluster and click **Job Management** in the **Operation** column. On the displayed page, click the **Links** tab and then **Create Link**.

Figure 8-23 Selecting a connector

Step 2 Select **RDS for MySQL** and click **Next** to set the link parameters.

Figure 8-24 Creating a MySQL link



Click **Show Advanced Attributes** to view more optional parameters. For details, see **Link to an RDS for MySQL/MySQL Database**. Retain the default values of the optional parameters and configure the mandatory parameters according to **Table 8-6**.

Table 8-6 MySQL link parameters

Parameter	Description	Example Value
Name	Unique link name	mysqllink
Database Server	IP address or domain name of the MySQL database server	-

Parameter	Description	Example Value
Port	MySQL database port	3306
Database Name	Name of the MySQL database	sqoop
Username	User who has the read, write, and delete permissions on the MySQL database	admin
Password	Password of the user	-
Use Local API	Whether to use the local API of the database for acceleration. (The system attempts to enable the **local_infile** system variable of the MySQL database.)	Yes
Use Agent	Whether to extract data from the data source through an agent	No
local_infile Character Set	When using local_infile to import data to MySQL, you can configure the encoding format.	utf8
Driver Version	Before connecting CDM to a relational database, you need to upload the JDK 8 .jar driver of the relational database. Download the MySQL driver 5.1.48 from **https://downloads.mysql.com/archives/c-j/**, obtain **mysql-connector-java-5.1.48.jar**, and upload it.	-

Step 3 Click **Save**. The **Link Management** page is displayed.

NOTE

If an error occurs during the saving, the security settings of the MySQL database are incorrect. In this case, you need to enable the EIP of the CDM cluster to access the MySQL database.

----End

Creating a DWS Link

Step 1 On the **Cluster Management** page, locate a cluster and click **Job Management** in the **Operation** column. On the displayed page, click the **Links** tab and then **Create Link**.

Figure 8-25 Selecting a connector type

Step 2 Select **Data Warehouse Service** and click **Next** to configure the DWS link parameters. Set the mandatory parameters listed in **Table 8-7** and retain the default values for the optional parameters.

Table 8-7 DWS link parameters

Parameter	Description	Example Value
Name	Enter a unique link name.	dwslink
Database Server	IP address or domain name of the DWS database	192.168.0.3
Port	DWS database port	8000
Database Name	Name of the DWS database	db_demo
Username	User who has the read, write, and delete permissions on the DWS database	dbadmin
Password	Password of the user	-
Use Agent	Whether to extract data from the data source through an agent	Yes
Agent	Click **Select** and select the created agent.	-
Import Mode	**COPY**: Migrate the source data to the DWS management node and then copy the data to DataNodes. To access DWS through the Internet, select **COPY**.	COPY

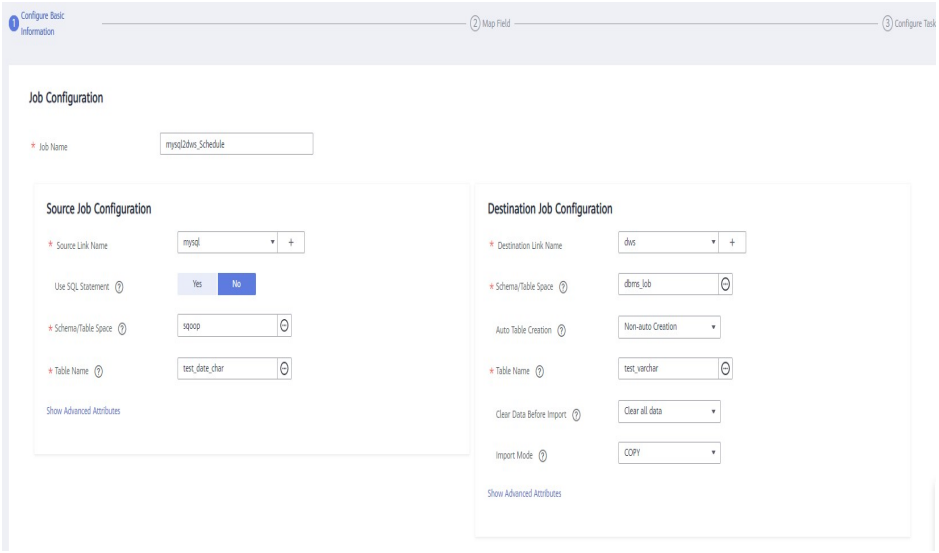
Step 3 Click **Save**.

----End

Creating a Migration Job

Step 1 Choose **Table/File Migration** > **Create Job** to create a job for exporting data from the MySQL database to DWS.

Figure 8-26 Creating a job for migrating data from MySQL to DWS



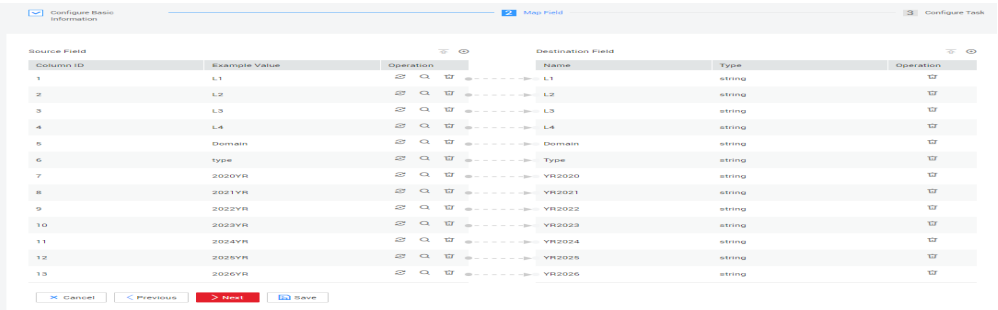
- **Job Name**: Enter a unique name.
- **Source Job Configuration**
  - **Source Link Name**: Select the **mysqllink** created in **Creating a MySQL Link**.
  - **Use SQL Statement**: Select **No**.
  - **Schema/Tablespace**: name of the schema or tablespace from which data is to be extracted
  - **Table Name**: name of the table from which data is to be extracted
  - Retain the default values of other optional parameters.
- **Destination Job Configuration**
  - **Destination Link Name**: Select the **dwslink** created in **Creating a DWS Link**.
  - **Schema/Tablespace**: Select the DWS database to which data is to be written.
  - **Auto Table Creation**: This parameter is displayed only when both the migration source and destination are relational databases.
  - **Table Name**: Name of the table to which data is to be written. You can enter a table name that does not exist. CDM automatically creates the table in DWS.
  - **isCompress**: whether to compress data. If you select **Yes**, high-level compression will be performed. CDM applies to compression scenarios where the I/O read/write volume is large and the CPU is sufficient (the computing load is relatively low). For more compression levels, see **Compression Levels**.

- **Orientation**: You can create row- or column-store tables as needed. Generally, if a table contains many columns (called a wide table) and its query involves only a few columns, column storage is recommended. If a table contains only a few columns and a query includes most of the fields, row storage is recommended.
- **Extend char length**: If the data encoding formats of the migration source and destination are different, the character length of the automatic table creation may be insufficient. If you select **Yes** for this parameter, the character length will be increased by three times during automatic table creation.
- **Clear Data Before Import**: whether to clear data in the destination table before the migration task starts.

Step 2 Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields, as shown in **Figure 8-27**.

- If the field mapping is incorrect, you can drag the fields to adjust the mapping.
- The expressions in CDM support field conversion of common character strings, dates, and values. For details, see **Converting Fields**.

Figure 8-27 Table-to-table field mapping



Step 3 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.
- **Schedule Execution**: Enable it if you need to configure scheduled jobs. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. You can increase the value of this parameter to improve migration efficiency.
- **Write Dirty Data**: Dirty data may be generated during data migration between tables. You are advised to select **Yes**.
- **Delete Job After Completion**: Retain the default value **Do not delete**.

Step 4 Click **Save and Run**. The **Job Management** page is displayed, on which you can view the job execution progress and result.

Step 5 After the job is successfully executed, in the **Operation** column of the job, click **Historical Record** to view the job's historical execution records and read/write statistics.

On the **Historical Record** page, click **Log** to view the job logs.

----End

8.6 Migrating an Entire MySQL Database to RDS

Scenario

This section describes how to migrate the entire on-premises MySQL database to RDS using the CDM's entire DB migration function.

Currently, CDM can migrate the entire on-premises MySQL database to RDS for MySQL, RDS for PostgreSQL, or RDS for SQL Server. The following describes how to migrate the entire database to RDS. The procedure is as follows:

1. **Creating a CDM Cluster and Binding an EIP to the Cluster**
2. **Creating a MySQL Link**
3. **Creating an RDS Link**
4. **Creating an Entire DB Migration Job**

Prerequisites

- You have sufficient EIP quota.
- You have obtained an RDS database instance and the database engine of this instance is MySQL.
- The on-premises MySQL database can be accessed through the public network. If the MySQL database is deployed on an on-premises data center or a third-party cloud, ensure that an IP address that can be accessed from the public network has been configured for the MySQL database, or the VPN or Direct Connect between the on-premises data center and the cloud service platform has been established.
- You have obtained the IP addresses, names, usernames, and passwords of the on-premises MySQL database and RDS for MySQL.
- You have uploaded the MySQL database driver on the **Job Management** > **Links** > **Driver Management** page.

Creating a CDM Cluster and Binding an EIP to the Cluster

Step 1 If CDM is used an independent service, create a CDM cluster by following the instructions in **Creating a CDM Cluster**. If CDM is used as a module of DataArts Studio, create a CDM cluster by following the instructions in **Creating a CDM Cluster**.

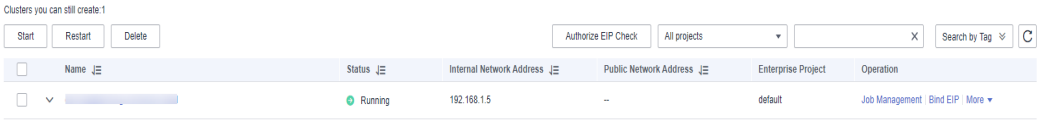
The key configurations are as follows:

- The flavor of the CDM cluster is selected based on the amount of data to be migrated. Generally, cdm.medium meets the requirements for most migration scenarios.

- The CDM cluster and the RDS for MySQL instance must be in the same VPC. In addition, it is recommended that the CDM cluster be in the same subnet and security group as the RDS for MySQL instance.
- If the same subnet and security group cannot be used for security purposes, ensure that a security group rule has been configured to allow the CDM cluster to access the RDS for MySQL instance.

Step 2 After the CDM cluster is created, on the **Cluster Management** page, click **Bind EIP** in the **Operation** column to bind an EIP to the cluster. The CDM cluster uses the EIP to access the on-premises MySQL database.

Figure 8-28 Cluster list



NOTE

If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

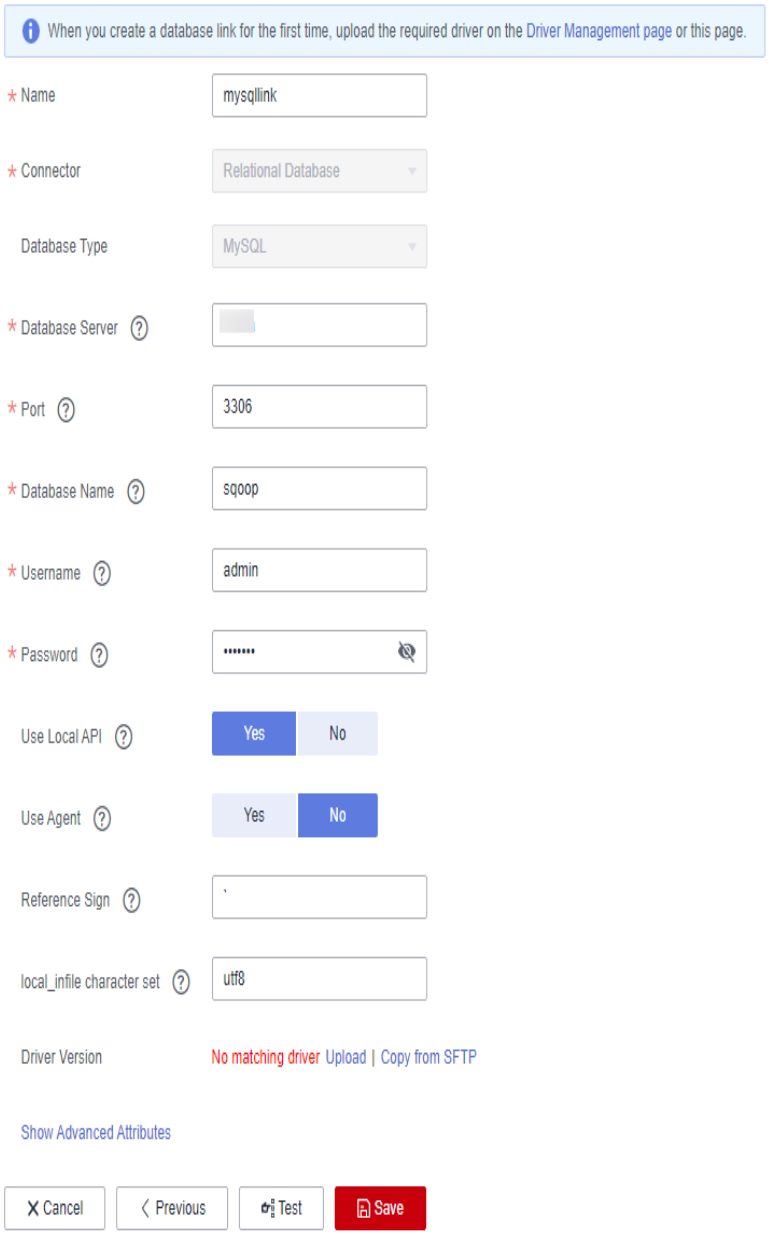
Creating a MySQL Link

Step 1 On the **Cluster Management** page, locate a cluster and click **Job Management** in the **Operation** column. On the displayed page, click the **Links** tab and then **Create Link**.

Figure 8-29 Selecting a connector

Step 2 Select **RDS for MySQL** and click **Next** to set the link parameters.

Figure 8-30 Creating a MySQL link



Click **Show Advanced Attributes** to view more optional parameters. For details, see **Link to an RDS for MySQL/MySQL Database**. Retain the default values of the optional parameters and configure the mandatory parameters according to **Table 8-8**.

Table 8-8 MySQL link parameters

Parameter	Description	Example Value
Name	Unique link name	mysqllink
Database Server	IP address or domain name of the MySQL database server	-

Parameter	Description	Example Value
Port	MySQL database port	3306
Database Name	Name of the MySQL database	sqoop
Username	User who has the read, write, and delete permissions on the MySQL database	admin
Password	Password of the user	-
Use Local API	Whether to use the local API of the database for acceleration. (The system attempts to enable the **local_infile** system variable of the MySQL database.)	Yes
Use Agent	Whether to extract data from the data source through an agent	No
local_infile Character Set	When using local_infile to import data to MySQL, you can configure the encoding format.	utf8
Driver Version	Before connecting CDM to a relational database, you need to upload the JDK 8 .jar driver of the relational database. Download the MySQL driver 5.1.48 from **https://downloads.mysql.com/archives/c-j/**, obtain **mysql-connector-java-5.1.48.jar**, and upload it.	-

Step 3 Click **Save**. The **Link Management** page is displayed.

NOTE

If an error occurs during the saving, the security settings of the MySQL database are incorrect. In this case, you need to enable the EIP of the CDM cluster to access the MySQL database.

----End

Creating an RDS Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 8-31 Selecting a connector type

Step 2 Select **RDS for MySQL** and click **Next** to configure parameters for the RDS for MySQL link.

- **Name**: Enter a custom link name, for example, **rds_link**.
- **Database Server** and **Port**: Enter the address information about the RDS for MySQL database.
- **Database Name**: Enter the name of the RDS for MySQL database.
- **Username** and **Password**: Enter the username and password used for logging in to the database.

NOTE

- During RDS link creation, if **Use Local API** in **Show Advanced Attributes** is set to **Yes**, you can use the LOAD DATA function provided by MySQL to speed up data import.
- The LOAD DATA function is disabled by default on RDS for MySQL, so you need to modify the parameter group of the MySQL instance and set **local_infile** to **ON** to enable this function.
- If the **local_infile** parameter group cannot be edited, it is the default parameter group. You need to create a parameter group and modify its value, and apply it to the MySQL instance of RDS.

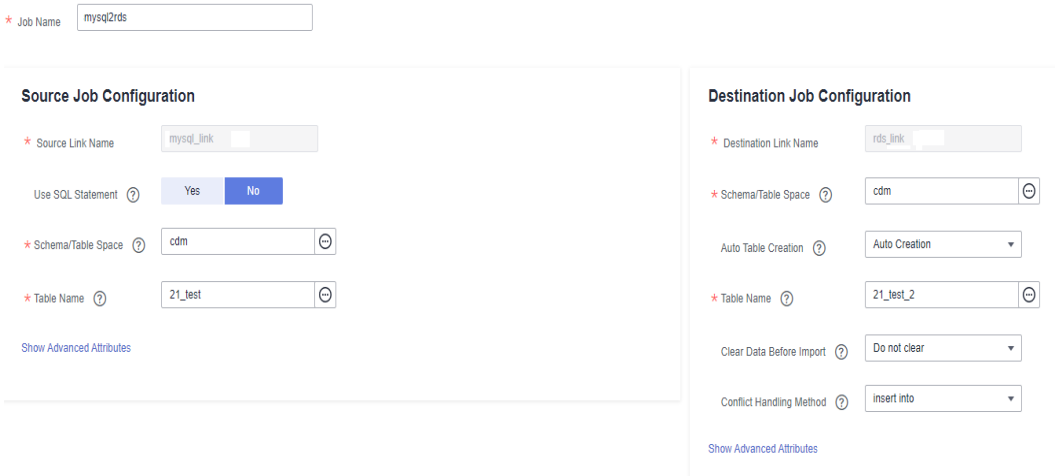
Step 3 Click **Save**. The **Link Management** page is displayed.

----End

Creating an Entire DB Migration Job

Step 1 After the two links are created, choose **Entire DB Migration** > **Create Job** to create a migration job. See **Figure 8-32**.

Figure 8-32 Creating an entire DB migration job



- **Job Name**: Enter a name for the entire DB migration job.
- **Source Job Configuration**
  - **Source Link Name**: Select the **mysqllink** created in **Creating a MySQL Link**.
  - **Schema/Tablespace**: Select the on-premises MySQL database from which data is to be exported.
- **Destination Job Configuration**
  - **Destination Link Name**: Select the **rds_link** link created in **Creating an RDS Link**.
  - **Schema/Tablespace**: Select the name of the RDS database to which data is to be imported.
  - **Auto Table Creation**: Select **Auto creation**, which indicates that CDM automatically creates tables in the RDS database when tables of the on-premises MySQL database do not exist in the RDS database.
  - **Clear Data Before Import**: Select **Yes**, which indicates that when a table with the same name as the table in the on-premises MySQL database exists in the RDS database, CDM clears data in the table on RDS.
  - **Constraint Conflict Handling**: Select **insert into**.
  - Retain the default values of the optional parameters in **Show Advanced Attributes**.

Step 2 Click **Next**. The page for selecting tables to be migrated is displayed. You can select all or part of tables to migrate.

Step 3 Click **Save and Run** and CDM immediately starts the entire DB migration job.

When the job starts running, a sub-job will be generated for each table. You can click the job name to view the sub-job list.

Step 4 In the **Operation** column of the job, click **Historical Record** to view the job's historical execution records and read/write statistics.

There are no logs for the entire DB migration job. However, the sub-jobs have logs. On the **Historical Record** page of the sub-jobs, click **Log** to view the job logs.

----End

8.7 Migrating Data from Oracle to CSS

Scenario

Cloud Search Service provides users with structured and unstructured data search, statistics, and report capabilities. This section describes how to use CDM to migrate data from the Oracle database to Cloud Search Service. The procedure is as follows:

1. **Creating a CDM Cluster and Binding an EIP to the Cluster**
2. **Creating a Cloud Search Service Link**
3. **Creating an Oracle Link**
4. **Creating a Migration Job**

Prerequisites

- You have subscribed to Cloud Search Service and obtained the IP address and port number of the Cloud Search Service cluster.
- You have obtained the IP address, name, username, and password of the Oracle database.
- If the Oracle database is deployed on an on-premises data center or a third-party cloud, ensure that an IP address that can be accessed from the public network has been configured for the Oracle database, or the VPN or Direct Connect between the on-premises data center and Huawei Cloud has been established.
- You have uploaded the Oracle database driver on the **Job Management** > **Links** > **Driver Management** page.

Creating a CDM Cluster and Binding an EIP to the Cluster

Step 1 If CDM is used an independent service, create a CDM cluster by following the instructions in **Creating a CDM Cluster**. If CDM is used as a module of DataArts Studio, create a CDM cluster by following the instructions in **Creating a CDM Cluster**.

The key configurations are as follows:

- The flavor of the CDM cluster is selected based on the amount of data to be migrated. Generally, cdm.medium meets the requirements for most migration scenarios.
- The CDM and Cloud Search Service clusters must be in the same VPC. In addition, it is recommended that the CDM cluster be in the same subnet and security group as the Cloud Search Service cluster.
- If the same subnet and security group cannot be used for security purposes, ensure that a security group rule has been configured to allow the CDM cluster to access the Cloud Search Service cluster.

Step 2 After the CDM cluster is created, on the **Cluster Management** page, click **Bind EIP** in the **Operation** column to bind an EIP to the cluster. The CDM cluster uses the EIP to access the Oracle data source.

NOTE

If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

Creating a Cloud Search Service Link

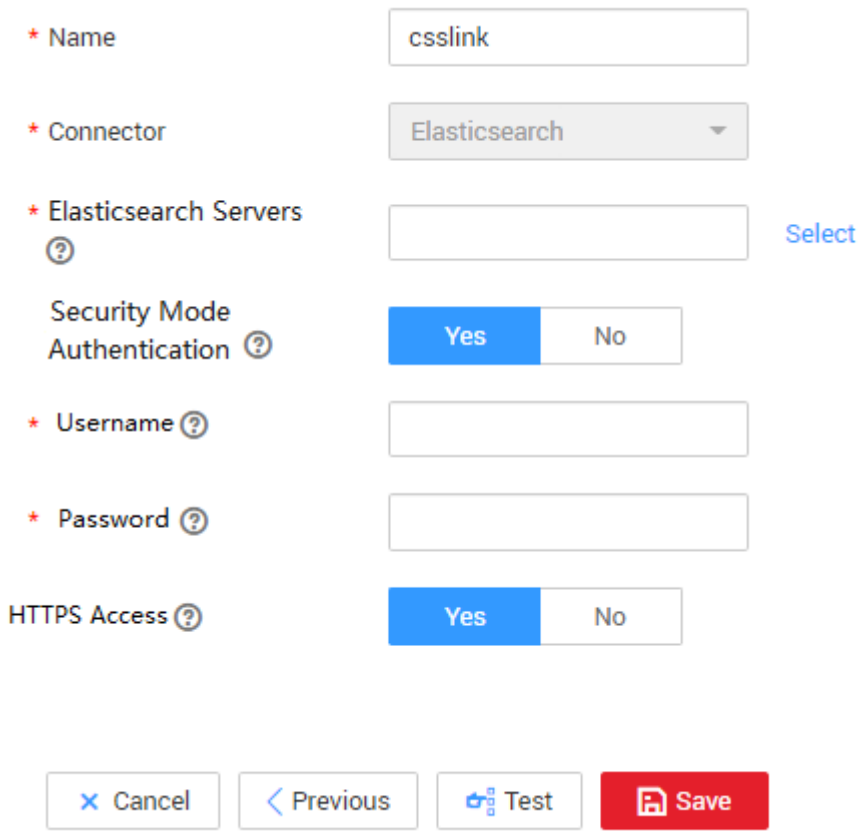
Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 8-33 Selecting a connector

Step 2 Select **Cloud Search Service** and click **Next**. On the page that is displayed, configure the CSS link parameters.

- **Name**: Enter a custom link name, for example, **csslink**.
- **Elasticsearch Server List**: Enter the IP address and port number of the Cloud Search Service cluster (cluster later than 5.*x*). The format is *ip:port*. Use semicolons to separate multiple addresses. For example, **192.168.0.1:9200;192.168.0.2:9200**.
- **Username** and **Password**: Enter the username and password used for logging in to the Cloud Search Service cluster. The user must have the read and write permissions on the database.

Figure 8-34 Creating a CSS link



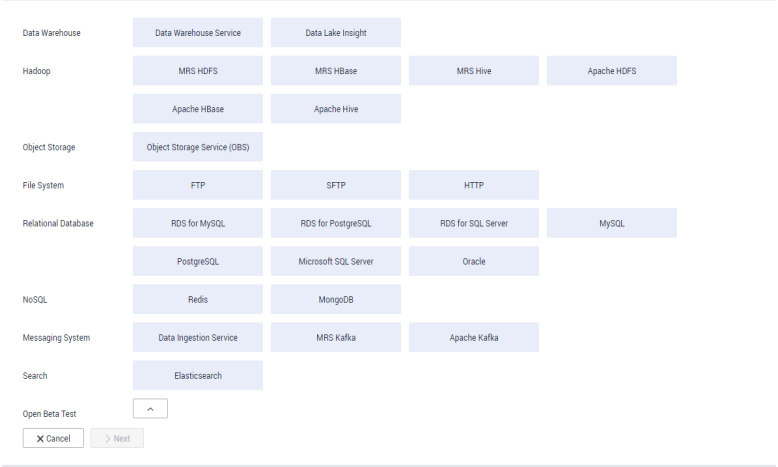
Step 3 Click **Save**. The **Link Management** page is displayed.

----End

Creating an Oracle Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 8-35 Selecting a connector type



Step 2 Select **Oracle** and click **Next** to configure parameters for the Oracle link.

- **Name**: Enter a custom link name, for example, **oracle_link**.
- **Database Server** and **Port**: Enter the address and port number of the Oracle server.
- **Database Name**: Enter the name of the Oracle database whose data is to be exported.
- **Username** and **Password**: Enter the username and password used for logging in to the Oracle database. The user must have the permission to read the Oracle metadata.

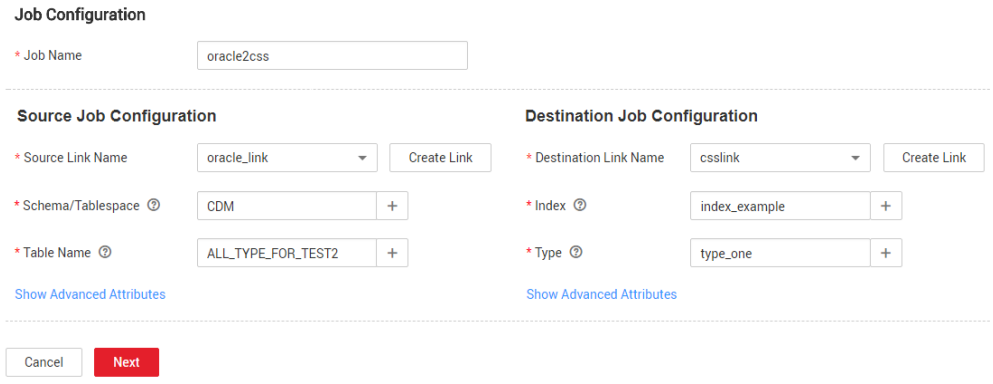
Step 3 Click **Save**. The **Link Management** page is displayed.

----End

Creating a Migration Job

Step 1 Choose **Table/File Migration** > **Create Job** to create a job for exporting data from the Oracle database to Cloud Search Service.

Figure 8-36 Creating a job for migrating data from Oracle to Cloud Search Service



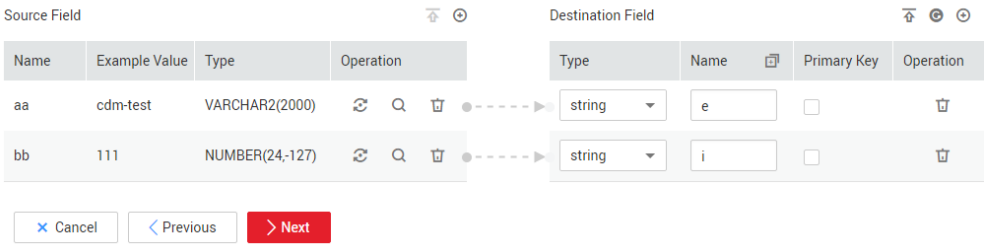
- **Job Name**: Enter a unique name.
- **Source Job Configuration**

  - **Source Link Name**: Select the **oracle_link** link created in **Creating an Oracle Link**.
  - **Schema/Tablespace**: Enter the name of the database whose data is to be migrated.
  - **Table Name**: Enter the name of the table to be migrated.
  - Retain the default values of the optional parameters in **Show Advanced Attributes**.
- **Destination Job Configuration**
  - **Destination Link Name**: Select the **csslink** link created in **Creating a Cloud Search Service Link**.
  - **Index**: Select the Elasticsearch index of the data to be written. You can also enter a new index. CDM automatically creates the index on Cloud Search Service.
  - **Type**: Select the Elasticsearch type of the data to be written. You can enter a new type. CDM automatically creates a type at the migration destination.
  - Retain the default values of the optional parameters in **Show Advanced Attributes**.

Step 2 Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields. See **Figure 8-37**.

- 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 supports field conversion during the migration. For details, see **Converting Fields**.

Figure 8-37 Field mapping of Cloud Search Service



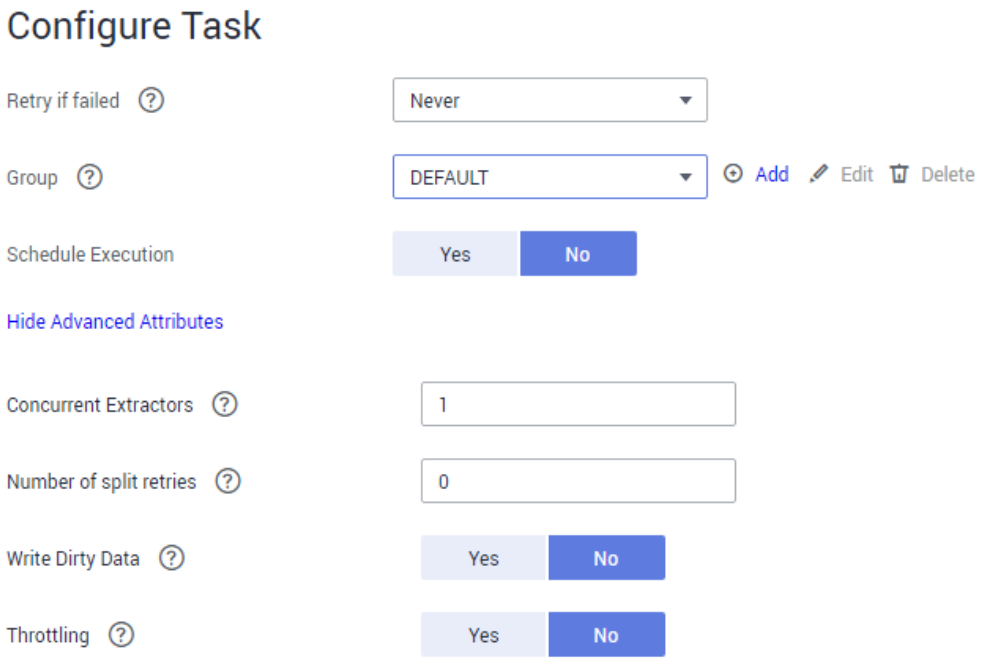
Step 3 Click **Next** and set task parameters. Generally, retain the default values of all parameters.

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

- **Retry If Failed**: Determine whether to automatically retry the job if it fails. 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.
- **Schedule Execution**: Determine whether to automatically execute the job at a scheduled time. Retain the default value **No** in this example.

- **Concurrent Extractors**: Enter the number of concurrent extractors. An appropriate value improves migration efficiency. For details, see **Performance Tuning**. 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 for future viewing. Before writing dirty data, create an OBS link on the CDM console. Retain the default value **No** so that dirty data is not recorded.

Figure 8-38 Configuring the task



Step 4 Click **Save and Run**. The **Job Management** page is displayed, on which you can view the job execution progress and result.

Step 5 After the job is successfully executed, in the **Operation** column of the job, click **Historical Record** to view the job's historical execution records and read/write statistics.

On the **Historical Record** page, click **Log** to view the job logs.

----End

8.8 Migrating Data from Oracle to DWS

Scenario

CDM supports table-to-table migration. This section describes how to use CDM to migrate data from Oracle to Data Warehouse Service (DWS). The procedure is as follows:

1. **Creating a CDM Cluster and Binding an EIP to the Cluster**
2. **Creating an Oracle Link**

3. **Creating a DWS Link**
4. **Creating a Migration Job**

Prerequisites

- You have obtained a DWS cluster and the IP address, port number, database name, username, and password for connecting to the DWS database. In addition, you must have the read, write, and delete permissions on the DWS database.
- You have obtained the IP address, name, username, and password of the Oracle database.
- If the Oracle database is deployed on an on-premises data center or a third-party cloud, ensure that an IP address that can be accessed from the public network has been configured for the Oracle database, or the VPN or Direct Connect between the on-premises data center and Huawei Cloud has been established.
- You have uploaded the Oracle database driver on the **Job Management** > **Links** > **Driver Management** page.

Creating a CDM Cluster and Binding an EIP to the Cluster

Step 1 If CDM is used an independent service, create a CDM cluster by following the instructions in **Creating a CDM Cluster**. If CDM is used as a module of DataArts Studio, create a CDM cluster by following the instructions in **Creating a CDM Cluster**.

The key configurations are as follows:

- The flavor of the CDM cluster is selected based on the amount of data to be migrated. Generally, cdm.medium meets the requirements for most migration scenarios.
- The VPC, subnet, and security group of the CDM cluster must be the same as those of the DWS cluster.
- If the same subnet and security group cannot be used, for security reasons, ensure that a security group rule has been configured to allow the CDM cluster to access the CSS cluster.

Step 2 After the CDM cluster is created, locate the row that contains the cluster and click **Bind EIP** in the **Operation** column. (CDM uses an EIP to access the Oracle data source.)

NOTE

If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

Creating an Oracle Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 8-39 Selecting a connector

Step 2 Select **Oracle** and click **Next** to configure parameters for the link.

Figure 8-40 Creating an Oracle link

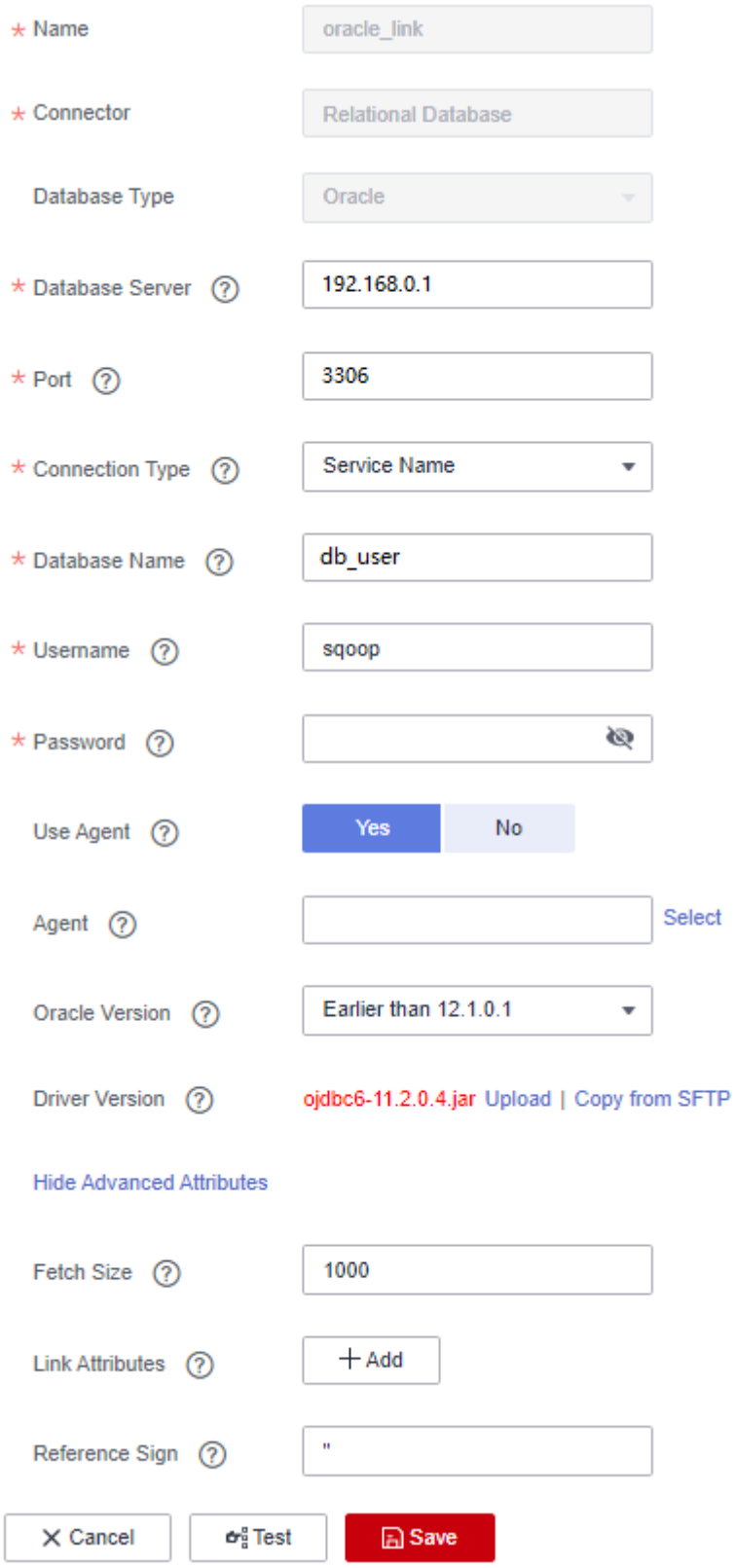


Table 8-9 Oracle link parameters

Parameter	Description	Example Value
Name	Enter a unique link name.	oracle_link
Database Server	Database server domain name or IP address	192.168.0.1
Port	Oracle database port	3306
Connection Type	Type of the Oracle database link	Service Name
Database Name	Name of the database to be connected	db_user
Username	User who has the read permission of the Oracle database	admin
Password	Password used for logging in to the Oracle database	-
Use Agent	Whether to extract data from the data source through an agent	Yes
Agent	Click **Select** and select the created agent.	-
Oracle Version	The latest version is used by default. If the version is incompatible, select another version.	Later than 12.1
Driver Version	A driver version that adapts to the Oracle database	-
Fetch Size	Number of rows obtained by each request	1000
Link Attributes	Custom attributes of the link	useCompression=true
Reference Sign	Delimiter used to separate referenced table names or column names This parameter is left blank by default.	'

Step 3 Click **Save**. The **Links** page is displayed.

----End

Creating a DWS Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 8-41 Selecting a connector type

Step 2 Select **Data Warehouse Service** and click **Next** to configure the DWS link parameters. Set the mandatory parameters listed in **Table 8-10** and retain the default values for the optional parameters.

Table 8-10 DWS link parameters

Parameter	Description	Example Value
Name	Enter a unique link name.	dwslink
Database Server	IP address or domain name of the DWS database	192.168.0.3
Port	DWS database port	8000
Database Name	Name of the DWS database	db_demo
Username	User who has the read, write, and delete permissions on the DWS database	dbadmin
Password	Password of the user	-
Use Agent	Whether to extract data from the data source through an agent	Yes
Agent	Click **Select** and select the created agent.	-
Import Mode	**COPY**: Migrate the source data to the DWS management node and then copy the data to DataNodes. To access DWS through the Internet, select **COPY**.	COPY

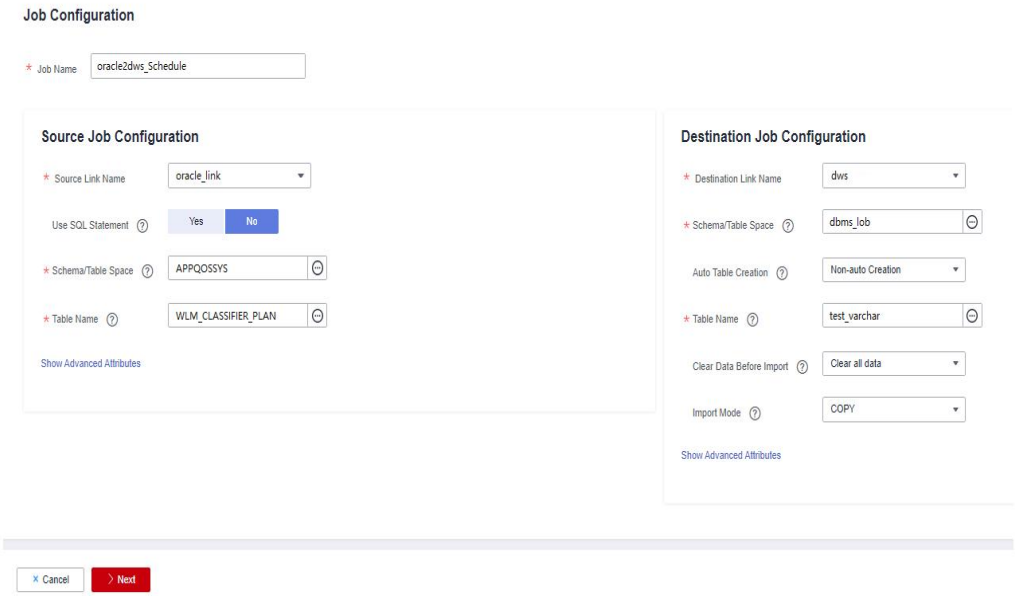
Step 3 Click **Save**.

----End

Creating a Migration Job

Step 1 Choose **Table/File Migration** > **Create Job** to create a job for exporting data from the Oracle database to DWS.

Figure 8-42 Creating a job for migrating data from Oracle to DWS



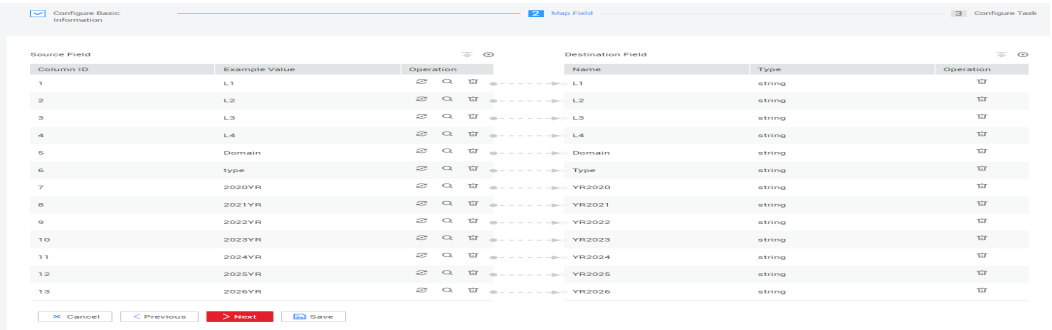
- **Job Name**: Enter a unique name.
- **Source Job Configuration**
  - **Source Link Name**: Select the **oracle_link** created in **Creating an Oracle Link**.
  - **Schema/Tablespace**: Enter the name of the database whose data is to be migrated.
  - **Table Name**: Enter the name of the table whose data is to be migrated.
  - Retain the default values of the optional parameters in **Show Advanced Attributes**.
- **Destination Job Configuration**
  - **Destination Link Name**: Select the **dwslink** created in **Creating a DWS Link**.
  - **Schema/Tablespace**: Select the DWS database to which data is to be written.
  - **Auto Table Creation**: This parameter is displayed only when both the migration source and destination are relational databases.
  - **Table Name**: Name of the table to which data is to be written. You can enter a table name that does not exist. CDM automatically creates the table in DWS.
  - **Orientation**: You can create row- or column-store tables as needed. Generally, if a table contains many columns (called a wide table) and its query involves only a few columns, column storage is recommended. If a table contains only a few columns and a query includes most of the fields, row storage is recommended.

- **Extend char length**: If the data encoding formats of the migration source and destination are different, the character length of the automatic table creation may be insufficient. If you select **Yes** for this parameter, the character length will be increased by three times during automatic table creation.
- **Clear Data Before Import**: whether to clear data in the destination table before the migration task starts.

Step 2 Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields, as shown in **Figure 8-43**.

- If the field mapping is incorrect, you can drag the fields to adjust the mapping.
- The expressions in CDM support field conversion of common character strings, dates, and values. For details, see **Converting Fields**.

Figure 8-43 Table-to-table field mapping



Step 3 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.
- **Schedule Execution**: Enable it if you need to configure scheduled jobs. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. You can increase the value of this parameter to improve migration efficiency.
- **Write Dirty Data**: Dirty data may be generated during data migration between tables. You are advised to select **Yes**.
- **Delete Job After Completion**: Retain the default value **Do not delete**.

Step 4 Click **Save and Run**. The **Job Management** page is displayed, on which you can view the job execution progress and result.

Step 5 After the job is successfully executed, in the **Operation** column of the job, click **Historical Record** to view the job's historical execution records and read/write statistics.

On the **Historical Record** page, click **Log** to view the job logs.

----End

NOTE

If the migration times out because writing data to the destination costs a long time, reduce the value of the **Fetch Size** parameter.

8.9 Migrating Data from OBS to CSS

Scenario

CDM supports data migration between cloud services. This section describes how to use CDM to migrate data from OBS to CSS. The procedure is as follows:

1. **Creating a CDM Cluster**
2. **Creating a Cloud Search Service Link**
3. **Creating an OBS Link**
4. **Creating a Migration Job**

Prerequisites

- You have obtained the domain name, port number, AK, and SK for accessing OBS.
- You have subscribed to Cloud Search Service and obtained the IP address and port number of the Cloud Search Service cluster.

Creating a CDM Cluster

If CDM is used an independent service, create a CDM cluster by following the instructions in **Creating a CDM Cluster**. If CDM is used as a module of DataArts Studio, create a CDM cluster by following the instructions in **Creating a CDM Cluster**.

The key configurations are as follows:

- The flavor of the CDM cluster is selected based on the amount of data to be migrated. Generally, cdm.medium meets the requirements for most migration scenarios.
- The CDM and Cloud Search Service clusters must be in the same VPC. In addition, it is recommended that the CDM cluster be in the same subnet and security group as the Cloud Search Service cluster.
- If the same subnet and security group cannot be used for security purposes, ensure that a security group rule has been configured to allow the CDM cluster to access the Cloud Search Service cluster.

Creating a Cloud Search Service Link

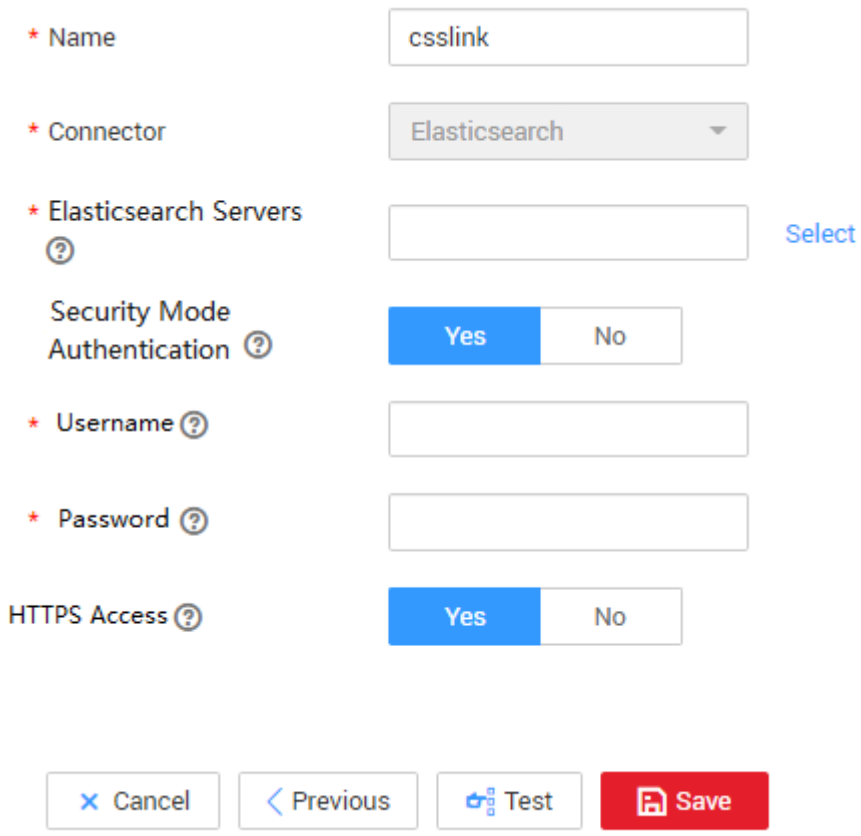
Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 8-44 Selecting a connector

Step 2 Select **Cloud Search Service** and click **Next**. On the page that is displayed, configure the CSS link parameters.

- **Name**: Enter a custom link name, for example, **csslink**.
- **Elasticsearch Server List**: Enter the IP address and port number of the Cloud Search Service cluster (cluster later than 5.*x*). The format is *ip:port*. Use semicolons to separate multiple addresses. For example, **192.168.0.1:9200;192.168.0.2:9200**.
- **Username** and **Password**: Enter the username and password used for logging in to the Cloud Search Service cluster. The user must have the read and write permissions on the database.

Figure 8-45 Creating a CSS link



Step 3 Click **Save**. The **Link Management** page is displayed.

----End

Creating an OBS Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 8-46 Selecting a connector type

Step 2 Select **Object Storage Service (OBS)** and click **Next** to configure parameters for the OBS link.

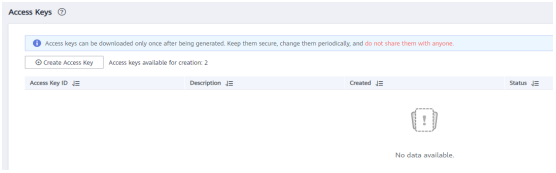
- **Name**: Enter a custom link name, for example, **obslink**.
- **OBS Server** and **Port**: Enter the actual OBS address information.
- **AK** and **SK**: Enter the AK and SK used for logging in to OBS.

  To obtain an access key, perform the following steps:

  a. Log in to the management console, move the cursor to the username in the upper right corner, and select **My Credentials** from the drop-down list.

  b. On the **My Credentials** page, choose **Access Keys**, and click **Create Access Key**. See **Figure 8-47**.

  Figure 8-47 Clicking Create Access Key

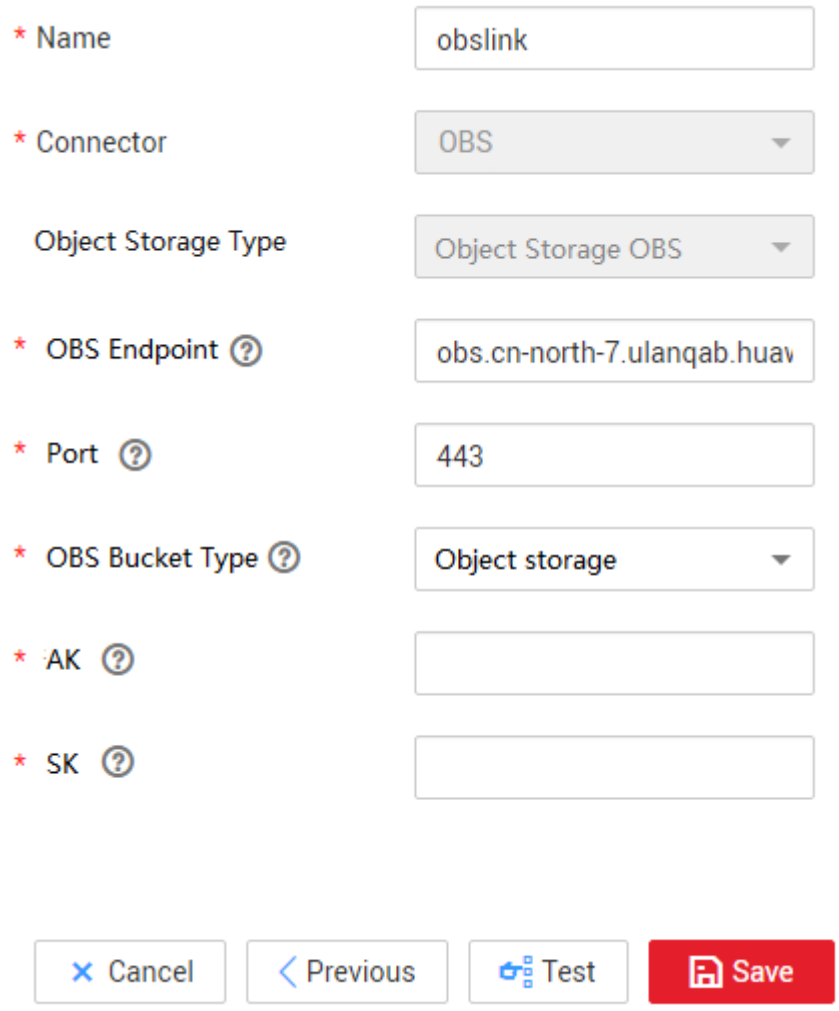
  

  c. Click **OK** and save the access key file as prompted. The access key file will be saved to your browser's configured download location. Open the **credentials.csv** file to view **Access Key Id** and **Secret Access Key**.

  NOTE

  - Only two access keys can be added for each user.
  - To ensure access key security, the access key is automatically downloaded only when it is generated for the first time and cannot be obtained from the management console later. Keep them properly.

Figure 8-48 Creating an OBS link



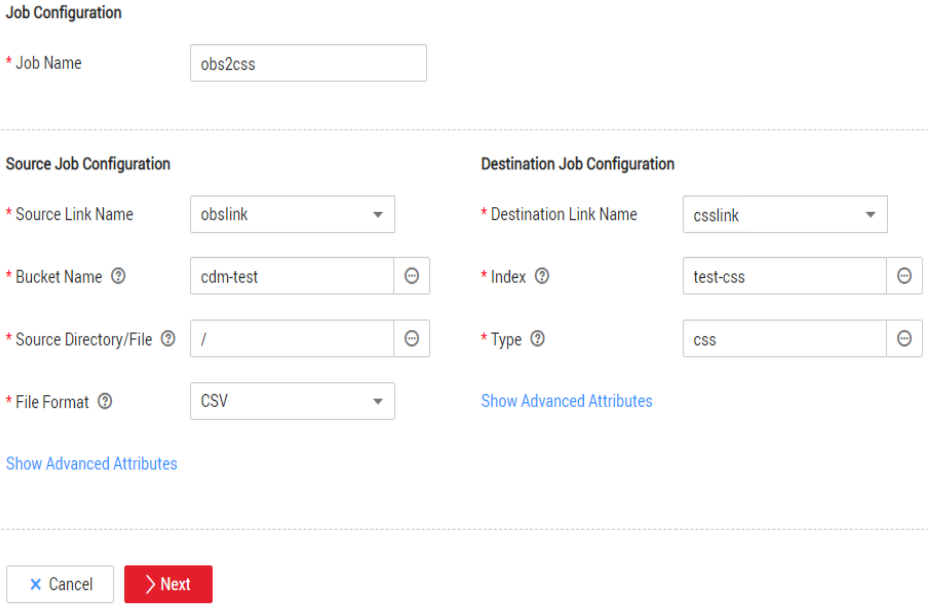
Step 3 Click **Save**. The **Link Management** page is displayed.

----End

Creating a Migration Job

Step 1 Choose **Table/File Migration** > **Create Job** to create a job for exporting data from OBS to Cloud Search Service.

Figure 8-49 Creating a job for migrating data from OBS to Cloud Search Service



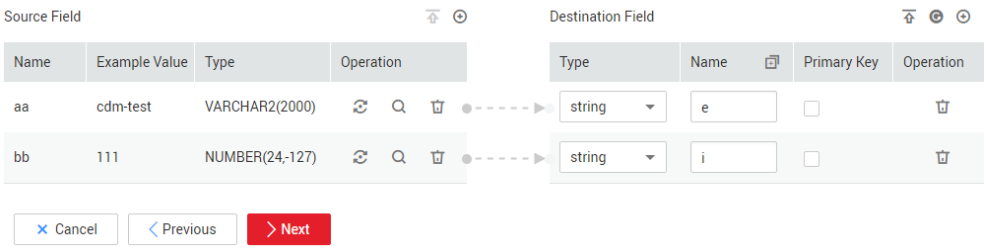
- **Job Name**: Enter a unique name.
- **Source Job Configuration**
  - **Source Link Name**: Select the **obslink** link created in **Creating an OBS Link**.
  - **Bucket Name**: Select the bucket from which the data will be migrated.
  - **Source Directory/File**: Set this parameter to the path of the data to be migrated. You can migrate all directories and files in the bucket.
  - **File Format**: Select **CSV** for migrating files to a data table.
  - Retain the default values of the optional parameters in **Show Advanced Attributes**.
- **Destination Job Configuration**
  - **Destination Link Name**: Select the **csslink** link created in **Creating a Cloud Search Service Link**.
  - **Index**: Select the Elasticsearch index of the data to be written. You can also enter a new index. CDM automatically creates the index on Cloud Search Service.
  - **Type**: Select the Elasticsearch type of the data to be written. You can enter a new type. CDM automatically creates a type at the migration destination.
  - Retain the default values of the optional parameters in **Show Advanced Attributes**.

Step 2 Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields. See **Figure 8-50**.

- 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 supports field conversion during the migration. For details, see **Converting Fields**.

Figure 8-50 Field mapping of Cloud Search Service

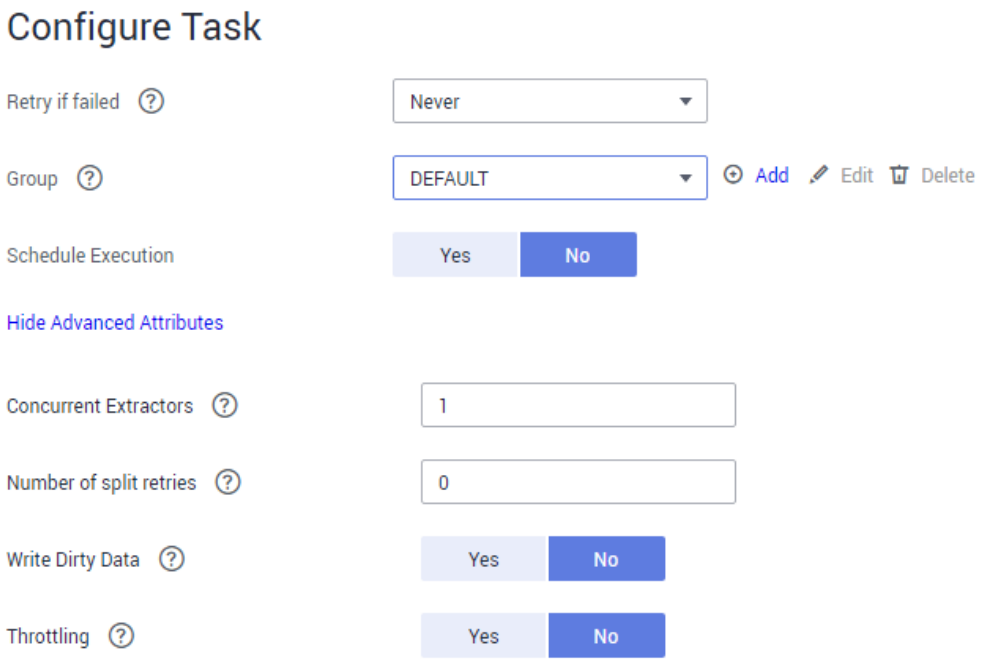


Step 3 Click **Next** and set task parameters. Generally, retain the default values of all parameters.

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

- **Retry If Failed**: Determine whether to automatically retry the job if it fails. 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.
- **Schedule Execution**: Determine whether to automatically execute the job at a scheduled time. Retain the default value **No** in this example.
- **Concurrent Extractors**: Enter the number of concurrent extractors. An appropriate value improves migration efficiency. For details, see **Performance Tuning**. 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 for future viewing. Before writing dirty data, create an OBS link on the CDM console. Retain the default value **No** so that dirty data is not recorded.

Figure 8-51 Configuring the task



Step 4 Click **Save and Run**. The **Job Management** page is displayed, on which you can view the job execution progress and result.

Step 5 After the job is successfully executed, in the **Operation** column of the job, click **Historical Record** to view the job's historical execution records and read/write statistics.

On the **Historical Record** page, click **Log** to view the job logs.

----End

8.10 Migrating Data from OBS to DLI

Scenario

DLI is a fully hosted big data query service. This section describes how to use CDM to migrate data from OBS to DLI. The procedure includes four steps:

1. **Creating a CDM Cluster**
2. **Creating a DLI Link**
3. **Creating an OBS Link**
4. **Creating a Migration Job**

Prerequisites

- You have enabled OBS and DLI and have the permissions to read data from OBS.
- You have created resource queues, databases, and tables on DLI.

Creating a CDM Cluster

If CDM is used an independent service, create a CDM cluster by following the instructions in **Creating a CDM Cluster**. If CDM is used as a module of DataArts Studio, create a CDM cluster by following the instructions in **Creating a CDM Cluster**.

In this scenario, if the CDM cluster is used only to migrate data from OBS to DLI and does not need to migrate data of other data sources, there is no special requirements on the VPC, subnet, and security group of the CDM cluster. You can specify them based on your needs. CDM accesses DLI and OBS through the intranet. The flavor of the CDM cluster is selected based on the amount of data to be migrated. Generally, cdm.medium meets the requirements for most migration scenarios.

Creating a DLI Link

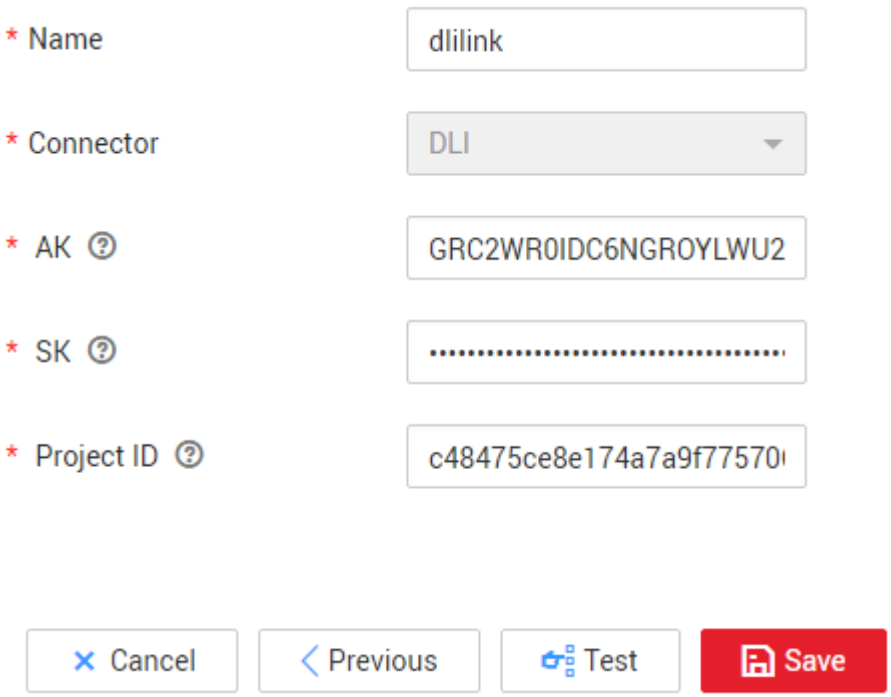
Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 8-52 Selecting a connector

Step 2 Select **Data Lake Insight**, click **Next**, and configure the DLI link parameters. See **Figure 8-53**.

- **Name**: Enter a custom link name, for example, **dlilink**.
- **AK** and **SK**: Enter the AK and SK used for accessing the DLI database.
- **Project ID**: Enter the project ID of the region to which DLI belongs.

Figure 8-53 Creating a DLI link



Step 3 Click **Save**. The **Link Management** page is displayed.

----End

Creating an OBS Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 8-54 Selecting a connector type

Step 2 Select **Object Storage Service (OBS)** and click **Next** to configure parameters for the OBS link.

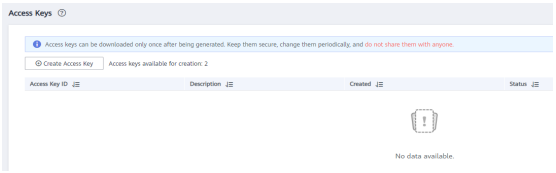
- **Name**: Enter a custom link name, for example, **obslink**.
- **OBS Server** and **Port**: Enter the actual OBS address information.
- **AK** and **SK**: Enter the AK and SK used for logging in to OBS.

  To obtain an access key, perform the following steps:

  a. Log in to the management console, move the cursor to the username in the upper right corner, and select **My Credentials** from the drop-down list.

  b. On the **My Credentials** page, choose **Access Keys**, and click **Create Access Key**. See **Figure 8-55**.

  Figure 8-55 Clicking Create Access Key

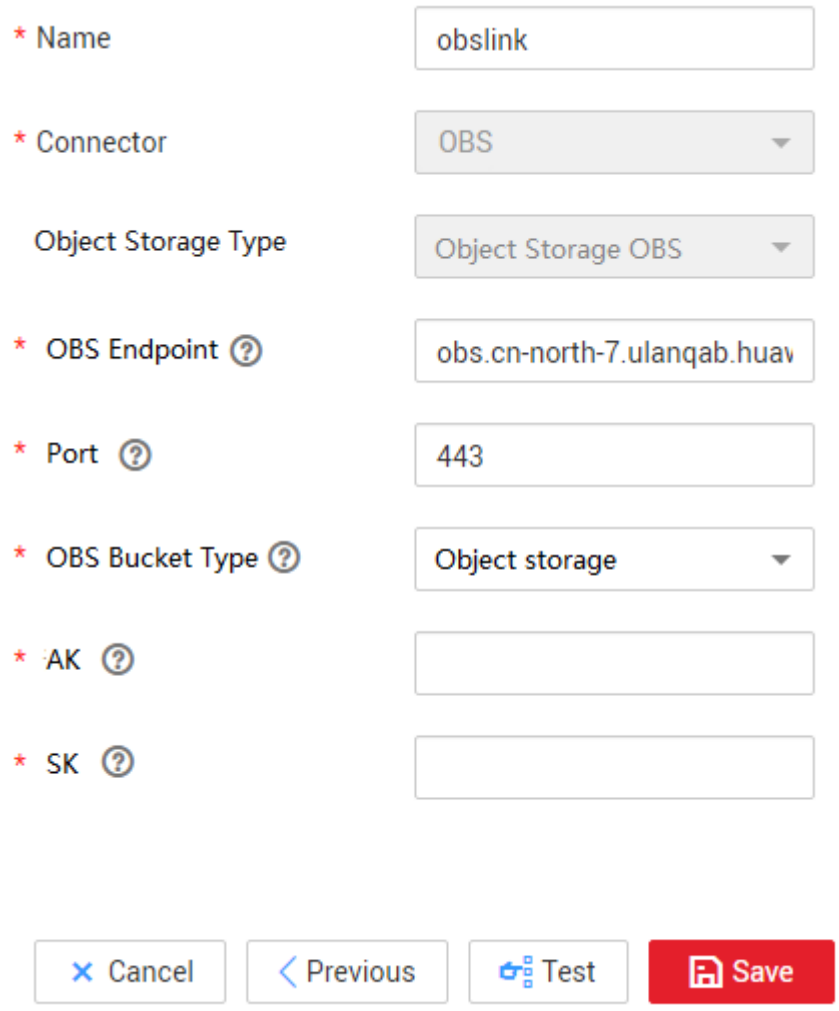
  

  c. Click **OK** and save the access key file as prompted. The access key file will be saved to your browser's configured download location. Open the **credentials.csv** file to view **Access Key Id** and **Secret Access Key**.

  NOTE

  - Only two access keys can be added for each user.
  - To ensure access key security, the access key is automatically downloaded only when it is generated for the first time and cannot be obtained from the management console later. Keep them properly.

Figure 8-56 Creating an OBS link



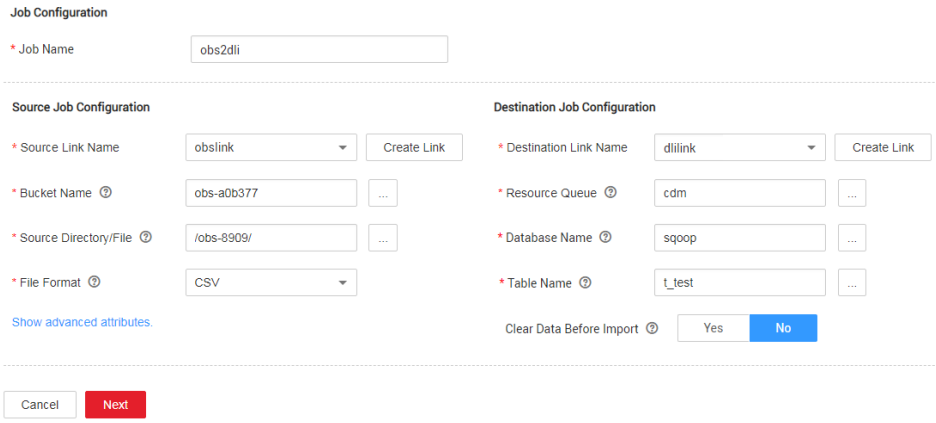
Step 3 Click **Save**. The **Link Management** page is displayed.

----End

Creating a Migration Job

Step 1 Choose **Table/File Migration** > **Create Job** to create a job for migrating data from OBS to DLI. See **Figure 8-57**.

Figure 8-57 Creating a job for migrating data from OBS to DLI



- **Job Name**: Enter a custom job name.
- **Source Link Name**: Select the **obslink** link created in **Creating an OBS Link**.
  - **Bucket Name**: Select the bucket from which the data is to be migrated.
  - **Source Directory/File**: Set this parameter to the path of the data to be migrated.
  - **File Format**: Select **CSV** or **JSON** for transferring files to a data table.
  - Retain the default values of the optional parameters in **Show Advanced Attributes**.
- **Destination Link Name**: Select the **dlilink** link created in **Creating a DLI Link**.
  - **Resource Queue**: Enter the resource queue to which the destination table belongs.
  - **Database Name**: Enter the name of the database to which data is to be written.
  - **Table Name**: Enter the name of the table to which data is to be written. CDM cannot automatically create tables on DLI. The table must be created on DLI in advance, and the field types and formats of the table must be consistent with those of the data to be migrated.
  - **Clear Before Importing Data**: Choose whether to clear data in the destination table before data import. In this example, retain the default value.

Step 2 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.
- CDM supports field conversion during the migration. For details, see **Converting Fields**.

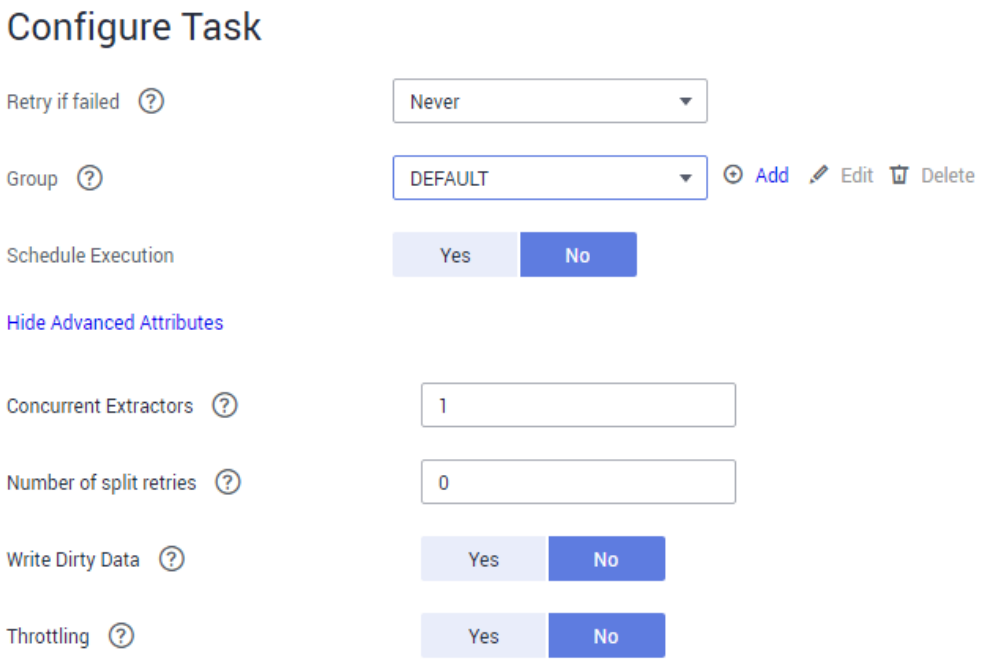
Step 3 Click **Next** and set task parameters. Generally, retain the default values of all parameters.

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

- **Retry If Failed**: Determine whether to automatically retry the job if it fails. 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.
- **Schedule Execution**: Determine whether to automatically execute the job at a scheduled time. Retain the default value **No** in this example.
- **Concurrent Extractors**: Enter the number of concurrent extractors. An appropriate value improves migration efficiency. For details, see **Performance Tuning**. 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 for future viewing. Before writing dirty data, create an OBS link on the CDM console. Retain the default value **No** so that dirty data is not recorded.

Figure 8-58 Configuring the task



Step 4 Click **Save and Run**. The **Job Management** page is displayed, on which you can view the job execution progress and result.

Step 5 After the job is successfully executed, in the **Operation** column of the job, click **Historical Record** to view the job's historical execution records and read/write statistics.

On the **Historical Record** page, click **Log** to view the job logs.

----End

8.11 Migrating Data from MRS HDFS to OBS

Scenario

CDM supports file-to-file data migration. This section describes how to migrate data from MRS HDFS to OBS. The process is as follows:

1. **Creating a CDM Cluster and Binding an EIP to the Cluster**
2. **Creating an MRS HDFS Link**
3. **Creating an OBS Link**
4. **Creating a Migration Job**

Prerequisites

- You have obtained the domain name, port number, AK, and SK for accessing OBS.
- You have purchased an MRS cluster.
- Your EIP quota is sufficient.

Creating a CDM Cluster and Binding an EIP to the Cluster

Step 1 If CDM is used an independent service, create a CDM cluster by following the instructions in **Creating a CDM Cluster**. If CDM is used as a module of DataArts Studio, create a CDM cluster by following the instructions in **Creating a CDM Cluster**.

The key configurations are as follows:

- The flavor of the CDM cluster is selected based on the amount of data to be migrated. Generally, cdm.medium meets the requirements for most migration scenarios.
- The VPC, subnet, and security group of the CDM cluster must be the same as those of the MRS cluster.

Step 2 After the CDM cluster is created, on the **Cluster Management** page, click **Bind EIP** in the **Operation** column to bind an EIP to the cluster. The CDM cluster uses the EIP to access MRS HDFS.

NOTE

If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

Creating an MRS HDFS Link

Step 1 On the **Cluster Management** page, locate a cluster and click **Job Management** in the **Operation** column. On the displayed page, click the **Links** tab and then **Create Link**.

Figure 8-59 Selecting a connector type

Step 2 Select **MRS HDFS** and click **Next** to configure parameters for the MRS HDFS link.

- **Name**: Enter a custom link name, for example, **mrs_hdfs_link**.
- **Manager IP**: IP address of MRS Manager. Click **Select** next to the **Manager IP** text box to select a created MRS cluster. CDM automatically fills in the authentication information.
- **Username**: If **Authentication Method** is set to **KERBEROS**, set the username and password for logging in to MRS Manager.

  If you need to create a snapshot when exporting a directory from HDFS, the user configured here must have the administrator permission on HDFS.
- **Password**: password for logging in to MRS Manager
- **Authentication Method**: authentication method for accessing MRS
- **Run Mode**: Select the running mode of the HDFS link.

----End

Creating an OBS Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 8-60 Selecting a connector type

Step 2 Select **Object Storage Service (OBS)** and click **Next** to configure parameters for the OBS link.

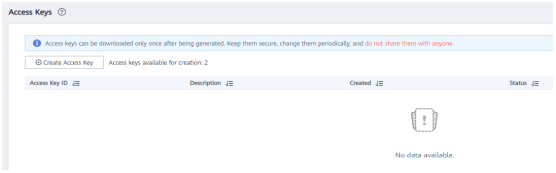
- **Name**: Enter a custom link name, for example, **obslink**.
- **OBS Server** and **Port**: Enter the actual OBS address information.
- **AK** and **SK**: Enter the AK and SK used for logging in to OBS.

  To obtain an access key, perform the following steps:

  a. Log in to the management console, move the cursor to the username in the upper right corner, and select **My Credentials** from the drop-down list.

  b. On the **My Credentials** page, choose **Access Keys**, and click **Create Access Key**. See **Figure 8-61**.

  Figure 8-61 Clicking Create Access Key

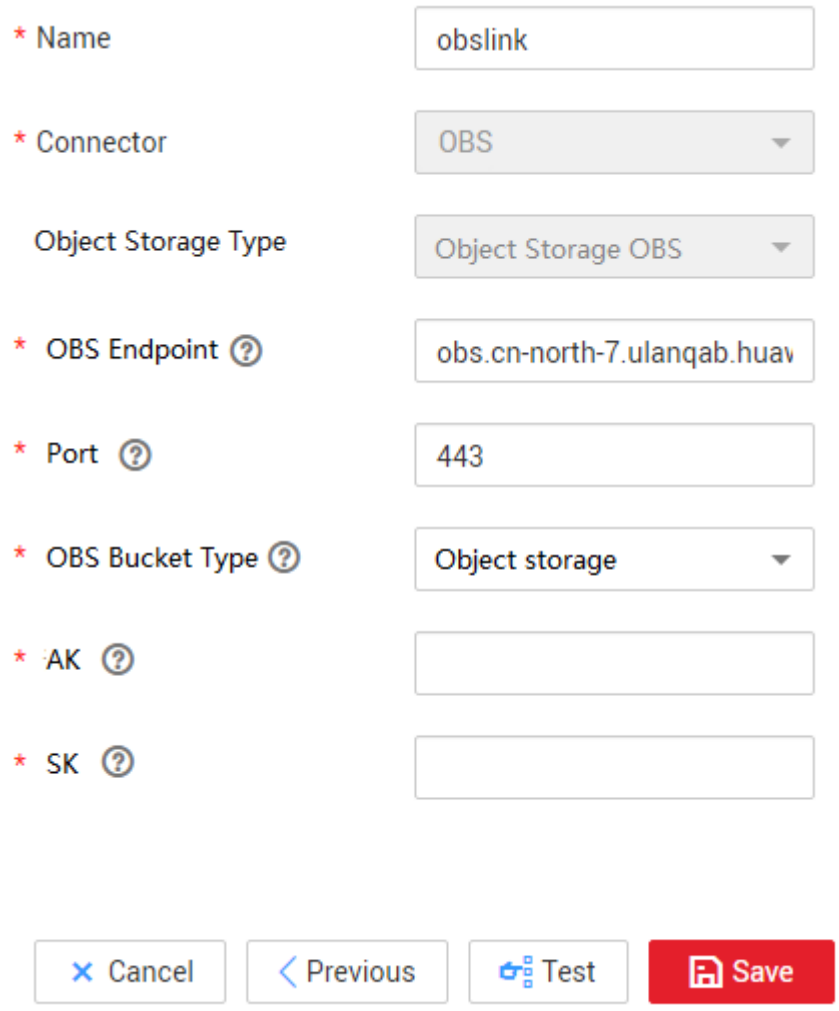
  

  c. Click **OK** and save the access key file as prompted. The access key file will be saved to your browser's configured download location. Open the **credentials.csv** file to view **Access Key Id** and **Secret Access Key**.

  NOTE

  - Only two access keys can be added for each user.
  - To ensure access key security, the access key is automatically downloaded only when it is generated for the first time and cannot be obtained from the management console later. Keep them properly.

Figure 8-62 Creating an OBS link



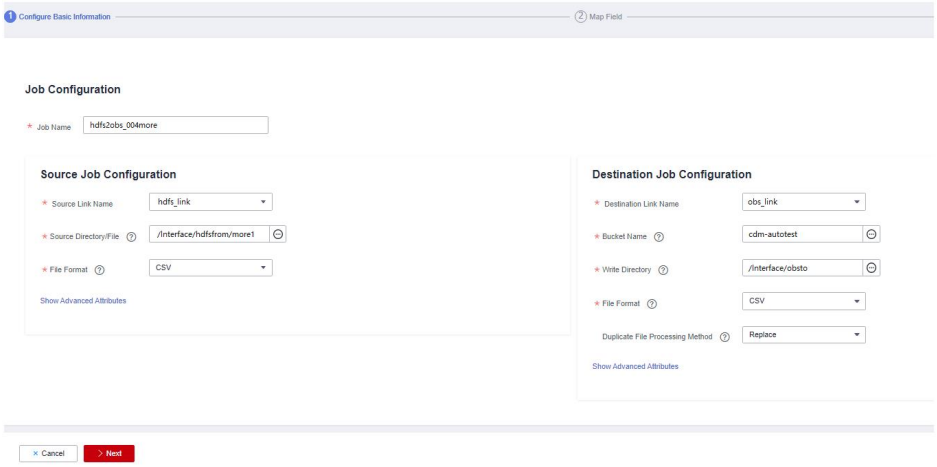
Step 3 Click **Save**. The **Link Management** page is displayed.

----End

Creating a Migration Job

Step 1 Choose **Table/File Migration** > **Create Job** to create a job for exporting data from the MRS HDFS database to OBS.

Figure 8-63 Creating a job for migrating data from MRS HDFS to OBS



- **Job Name**: Enter a unique name.
- **Source Job Configuration**
  - **Source Link Name**: Select the **hdfs_llink** created in **Creating an MRS HDFS Link**.
  - **Source Directory/File**: Enter the directory or file path of the data to be migrated.
  - **File Format**: Select the file format used for data transmission. Select **Binary**. If files are transferred without being parsed, the file format does not have to be **Binary**. This applies to file copy.
  - Retain the default values of other optional parameters.
- **Destination Job Configuration**
  - **Destination Link Name**: Select the **obs_link** created in **Creating an OBS Link**.
  - **Bucket Name**: Select the bucket from which the data will be migrated.
  - **Write Directory**: Enter the directory to which data is to be written on the OBS server.
  - **File Format**: Select **Binary**.
  - Retain the default values of the optional parameters in **Show Advanced Attributes**.

Step 2 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.
- The expressions in CDM support field conversion of common character strings, dates, and values. For details, see **Converting Fields**.

Step 3 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.
- **Schedule Execution**: Enable it if you need to configure scheduled jobs. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. CDM supports concurrent extraction of multiple files. Increasing the value of this parameter can improve migration efficiency.
- **Write Dirty Data**: Select **No**. The file-to-file migration is binary, and no dirty data will be generated.
- **Delete Job After Completion**: Retain the default value **Do not delete**. You can also set this parameter to **Delete** to prevent an accumulation of too many migration jobs.

Step 4 Click **Save and Run**. The **Job Management** page is displayed, on which you can view the job execution progress and result.

Step 5 After the job is successfully executed, in the **Operation** column of the job, click **Historical Record** to view the job's historical execution records and read/write statistics.

On the **Historical Record** page, click **Log** to view the job logs.

----End

8.12 Migrating the Entire Elasticsearch Database to CSS

Scenario

CSS provides users with structured and unstructured data search, statistics, and report capabilities. This section describes how to use CDM to migrate the entire Elasticsearch database to Cloud Search Service. The procedure is as follows:

1. **Creating a CDM Cluster and Binding an EIP to the Cluster**
2. **Creating a Cloud Search Service Link**
3. **Creating an Elasticsearch Link**
4. **Creating an Entire DB Migration Job**

Prerequisites

- You have sufficient EIP quota.
- You have subscribed to CSS and obtained the IP address and port number of the CSS cluster.
- You have obtained the IP address, port number, username, and password of the on-premises Elasticsearch database server.

  If the Elasticsearch server is deployed on an on-premises data center or a third-party cloud, ensure that an IP address that can be accessed from the public network has been configured for the Elasticsearch server, or the VPN or Direct Connect between the on-premises data center and HUAWEI CLOUD has been established.

Creating a CDM Cluster and Binding an EIP to the Cluster

Step 1 If CDM is used an independent service, create a CDM cluster by following the instructions in **Creating a CDM Cluster**. If CDM is used as a module of DataArts Studio, create a CDM cluster by following the instructions in **Creating a CDM Cluster**.

The key configurations are as follows:

- The flavor of the CDM cluster is selected based on the amount of data to be migrated. Generally, cdm.medium meets the requirements for most migration scenarios.
- The CDM and Cloud Search Service clusters must be in the same VPC. In addition, it is recommended that the CDM cluster be in the same subnet and security group as the Cloud Search Service cluster.
- If the same subnet and security group cannot be used for security purposes, ensure that a security group rule has been configured to allow the CDM cluster to access the Cloud Search Service cluster.

Step 2 After the CDM cluster is created, on the **Cluster Management** page, click **Bind EIP** in the **Operation** column to bind an EIP to the cluster. The CDM cluster uses the EIP to access the on-premises Elasticsearch.

NOTE

If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

Creating a Cloud Search Service Link

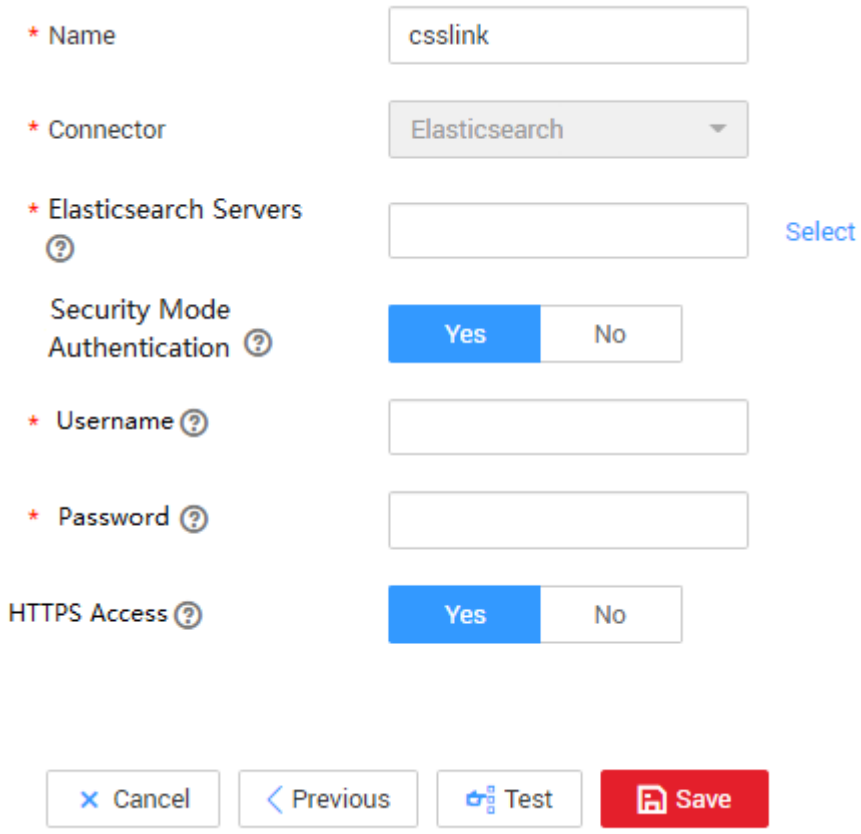
Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 8-64 Selecting a connector

Step 2 Select **Cloud Search Service** and click **Next**. On the page that is displayed, configure the CSS link parameters.

- **Name**: Enter a custom link name, for example, **csslink**.
- **Elasticsearch Server List**: Enter the IP address and port number of the Cloud Search Service cluster (cluster later than 5.*x*). The format is *ip:port*. Use semicolons to separate multiple addresses. For example, **192.168.0.1:9200;192.168.0.2:9200**.
- **Username** and **Password**: Enter the username and password used for logging in to the Cloud Search Service cluster. The user must have the read and write permissions on the database.

Figure 8-65 Creating a CSS link



Step 3 Click **Save**. The **Link Management** page is displayed.

----End

Creating an Elasticsearch Link

Step 1 On the **Cluster Management** page, locate a cluster and click **Job Management** in the **Operation** column. On the displayed page, click the **Links** tab and then **Create Link**.

Figure 8-66 Selecting a connector type

Step 2 Select **Elasticsearch** and click **Next** to configure parameters for the Elasticsearch link. The parameters are the same as those for the CSS link.

- **Name**: Enter a custom link name, for example, **es_link**.
- **Elasticsearch Server List**: Enter the IP address and port number of the on-premises Elasticsearch database. Use semicolons to separate multiple addresses.

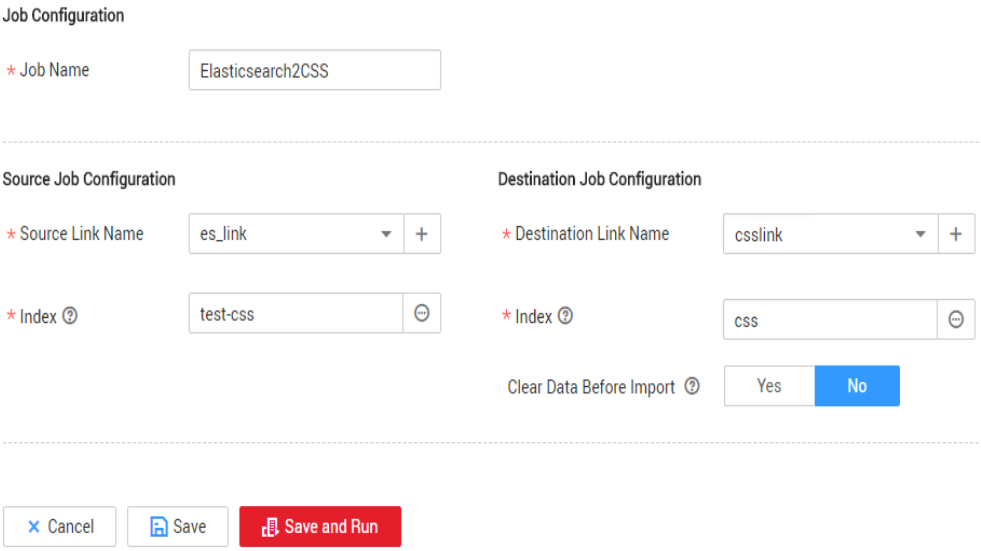
Step 3 Click **Save**. The **Link Management** page is displayed.

----End

Creating an Entire DB Migration Job

Step 1 Choose **Entire DB Migration** > **Create Job** to create an entire DB migration job.

Figure 8-67 Creating an entire DB migration job



- **Job Name**: Enter a unique name.

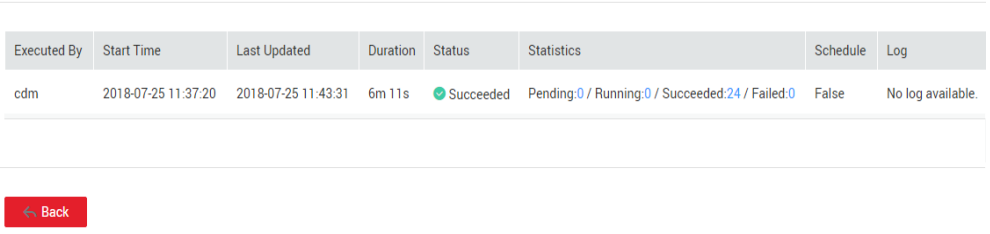
- **Source Job Configuration**
  - **Source Link Name**: Select the **es_link** link created in **Creating an Elasticsearch Link**.
  - **Index**: Click the icon next to the text box to select an index in the on-premises Elasticsearch database or manually enter an index name. The name can contain only lowercase letters. If multiple indexes need to be migrated at a time, set this parameter to a wildcard character. CDM migrates all indexes that meet the wildcard condition. For example, if this parameter is set to **cdm***, CDM migrates all indexes starting with **cdm**, such as **cdm01**, **cdmB3**, **cdm_45** and so on.
- **Destination Job Configuration**
  - **Destination Link Name**: Select the **csslink** link created in **Creating a Cloud Search Service Link**.
  - **Index**: Enter the index of the data to be written. You can select an existing index in Cloud Search Service or manually enter an index name that does not exist. The name can contain only lowercase letters. CDM automatically creates the index in Cloud Search Service. If multiple indexes are migrated at a time, this parameter cannot be configured. CDM automatically creates indexes at the migration destination.
  - **Clear Data Before Import**: If the selected index already exists in Cloud Search Service, you can choose whether to clear the data in the index before importing data. If you select **No**, the data is added to the index.

Step 2 Click **Save and Run**. The **Job Management** page is displayed, on which you can view the job execution progress and result.

A sub-job will be generated for each type in the on-premises Elasticsearch index for concurrent execution. You can click the job name to view the sub-job progress.

Step 3 After the job is successfully executed, in the **Operation** column of the job, click **Historical Record** to view the job's historical execution records, read/write statistics, and job logs (only the sub-jobs have job logs).

Figure 8-68 Historical Record



----End

