Cloud Data Migration

User Guide

Issue 18

Date 2020-10-23

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Contents

1 Permissions Management

1.1 Creating a User and Granting CDM Permissions

This chapter describes how to use **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.

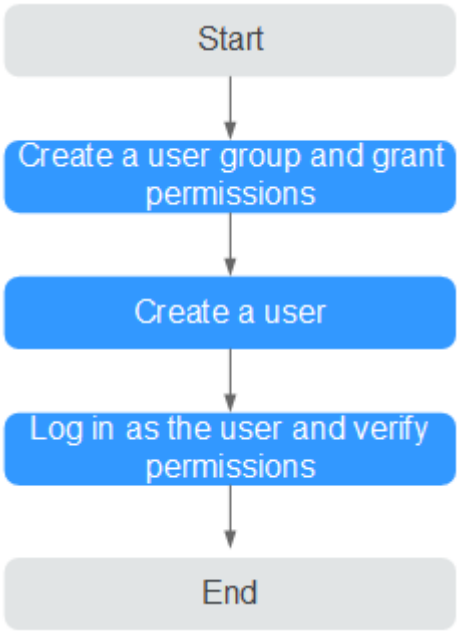
This section describes the procedure for granting permissions (see **Figure 1-1**).

Prerequisites

Learn about the permissions (see **Permissions Management**) supported by CDM and choose policies or roles according to your requirements. For the permissions of other services, see **System Permissions**.

Process Flow

Figure 1-1 Process of granting CDM permissions



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

   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"
```



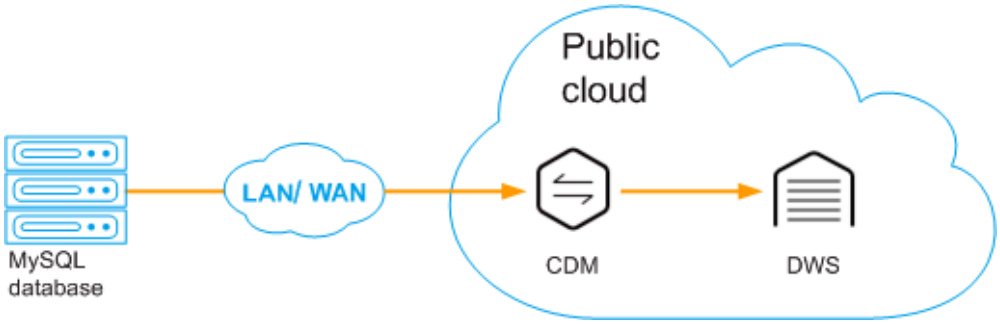
```
        }
    ]
}
```

2 Getting Started

2.1 Overview

This section describes how to use CDM to migrate the tables in the on-premises MySQL database to DWS, thereby helping you get familiar with CDM. **Figure 2-1** shows the specific scenario.

Figure 2-1 Migrating data from a local MySQL database to DWS



The procedure of using CDM is as follows:

1. **Creating a CDM Cluster**
2. **Creating Links**
3. **Creating and Executing a Job**
4. **Viewing Job Execution Result**

2.2 Step 1: Creating a Cluster

Scenario

This section describes how to create a CDM cluster to migrate data between an on-premises MySQL database and DWS.

Prerequisites

You have obtained the region, VPC, subnet, and security group of the data warehouse cluster.

Procedure

Step 1 Log in to the **CDM management console**.

Step 2 Click **Buy CDM Cluster**. The page for buying a CDM cluster is displayed. The following is a cluster configuration example:

- **Region**: Select the region where the CDM cluster resides. Resources in different regions cannot communicate with each other. The region must be the same as that of the data warehouse cluster.
- **AZ**: An AZ is a physical region where resources use independent power supply and networks. AZs are physically isolated but interconnected through an internal network. Select **AZ2**.
- **Name**: The cluster name must start with a letter and contains 4 to 64 characters consisting of letters, digits, hyphens (-), and underscores (_). It cannot contain special characters. For example, **cdm-aff1**.
- **Version**: Retain the default value.
- **Instance Type**: Select an instance flavor based on your service data volume.
  - **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 20 jobs can be executed concurrently. This flavor is well suited to migrating a single database table with 10 million pieces of data or more.
  - **cdm.xlarge**: 16 vCPUs and 32 GB of memory

    The maximum and assured bandwidths are 10 Gbit/s and 4 Gbit/s. Up to 100 jobs can be executed concurrently. This flavor is well suited to TB-level data migration requiring 10GE high-speed bandwidth.
  - **cdm.4xlarge**: 64 vCPUs and 128 GB of memory

    The maximum and assured bandwidths are both 30 Gbit/s. Up to 300 jobs can be executed concurrently.
- **VPC**: Select the VPC where DWS resides.
- **Subnet**: You are advised to use the same subnet as that of DWS.
- **Security Group**: You are advised to use the security group as that of DWS.

  You can select a subnet and security group that are different from those of DWS. In this case, configure the security group rules to allow the CDM cluster to properly access DWS.
- Retain the default values for other parameters.

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.

NOTE

Generally, it takes 10 to 20 minutes to create a cluster. If you create a CDM cluster for the first time, it takes only one minute to create it.

----End

2.3 Step 2: Creating Links

Description

Before migrating the local MySQL database to DWS, create two links:

1. MySQL link: used to connect to the on-premises MySQL database.
2. DWS link: used to connect to the DWS database.

CDM needs to access the on-premises data source. Therefore, before creating a link, bind an EIP to the CDM cluster.

Prerequisites

- Your on-premises MySQL database can be accessed using the public IP address.
- You have sufficient EIP quota.
- You have obtained the IP address, port number, database name, username, and password for connecting to the MySQL database. In addition, the user must have the read, write, and delete permissions on the MySQL database.
- You have a DWS instance and have obtained the IP address, port number, database name, username, and password for accessing DWS. Additionally, the account has the read, write, and delete permissions for the DWS database.

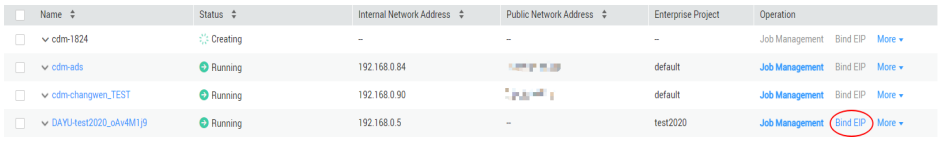
Creating a MySQL Link

Step 1 Log in to the **CDM management console**.

Step 2 In the left navigation pane, click **Cluster Management**. Locate the **cdm-aff1** cluster created in **Step 1: Creating a Cluster**.

Step 3 In the **Operation** column, click **Bind EIP**, and select and bind an EIP to the cluster.

Figure 2-2 Binding an EIP

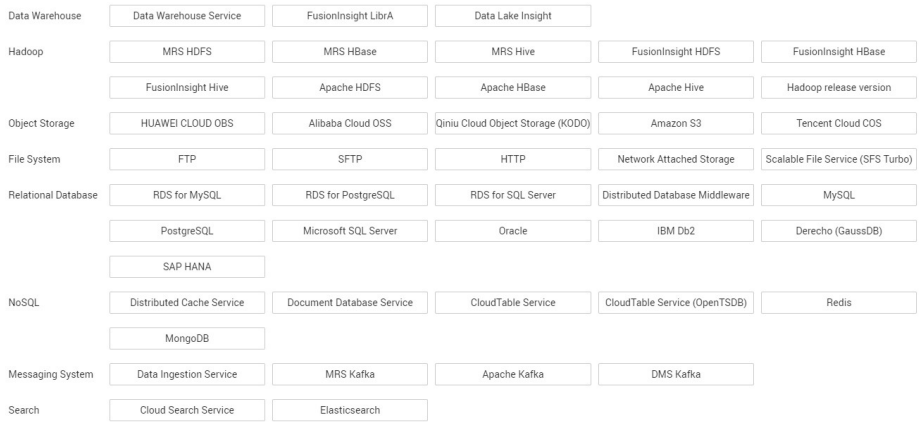


NOTE

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

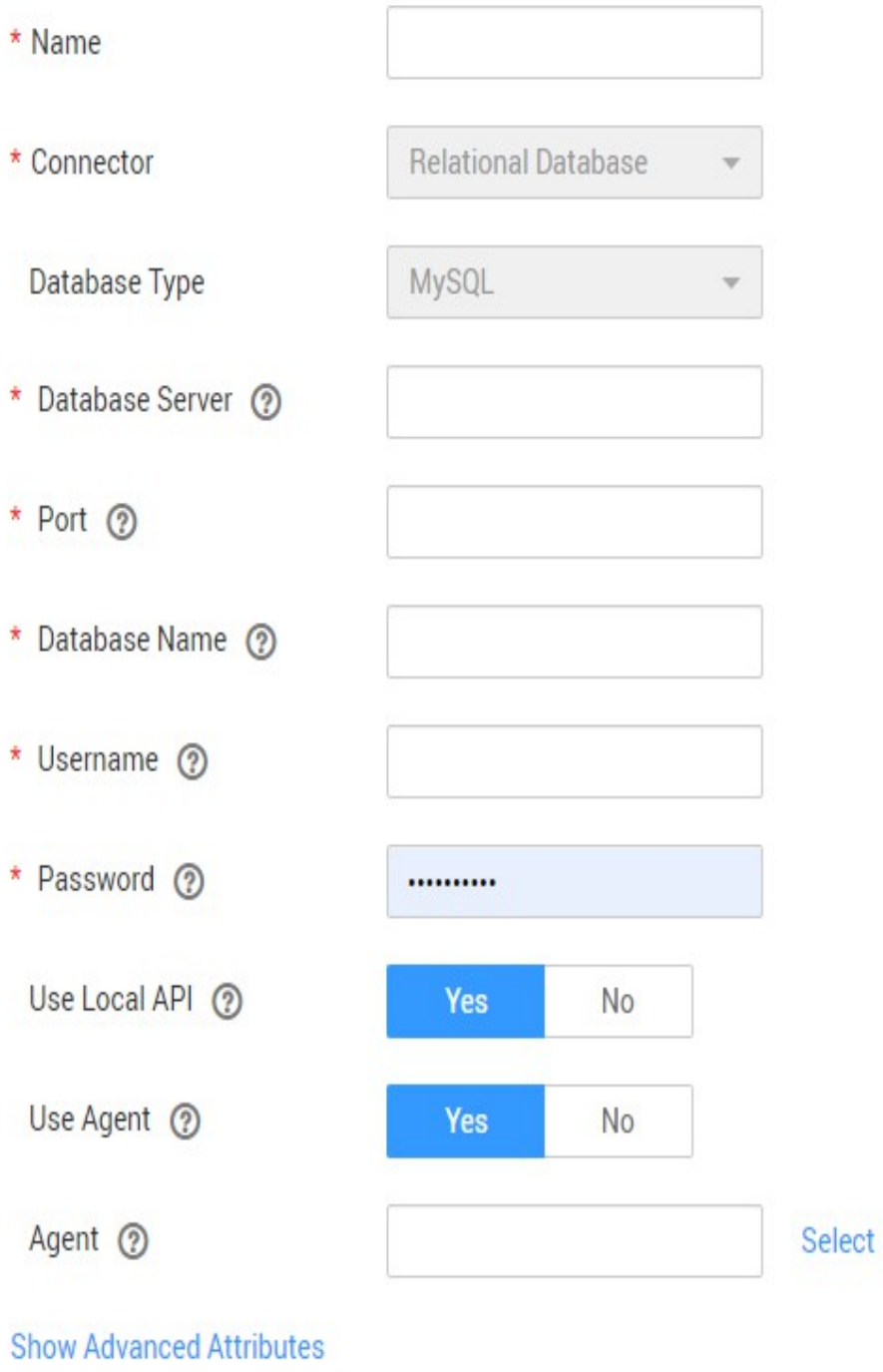
Step 4 Click **Job Management** in the **Operation** column of the CDM cluster. On the page that is displayed, choose **Link Management** > **Create Link**. The page for selecting a connector is displayed. See **Figure 2-3**.

Figure 2-3 Selecting a connector



Step 5 Select **MySQL** and click **Next**. On the page that is displayed, configure MySQL link parameters.

Figure 2-4 Creating a MySQL link



Click **Show Advanced Attributes** to display optional parameters. For details, see **Link to Relational Databases**. Retain the default values of the optional parameters and configure the mandatory parameters according to **Table 2-1**.

Table 2-1 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	192.168.0.1
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 Agent	Whether to extract data from the data source through an agent	Yes
Agent	Click **Select** and select the agent created in **Connecting to an Agent**.	-

Step 6 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 **Link Management** tab page, click **Create Link** and select **Data Warehouse Service** to create a DWS link.

Step 2 Click **Next**. The page for configuring the DWS link parameters is displayed. Configure the mandatory parameters according to **Table 2-2** and retain the default values of the optional parameters.

Table 2-2 DWS link parameters

Parameter	Description	Example Value
Name	Unique link name	dwslink
Database Server	IP address or domain name of the DWS database server	192.168.0.3
Port	DWS database port	8000
Database Name	Name of the DWS database	db_demo

Parameter	Description	Example Value
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 agent created in **Connecting to an 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

2.4 Step 3: Creating and Executing a Job

Scenario

This section describes how to create a table migration job to migrate data tables from an on-premises MySQL database to DWS.

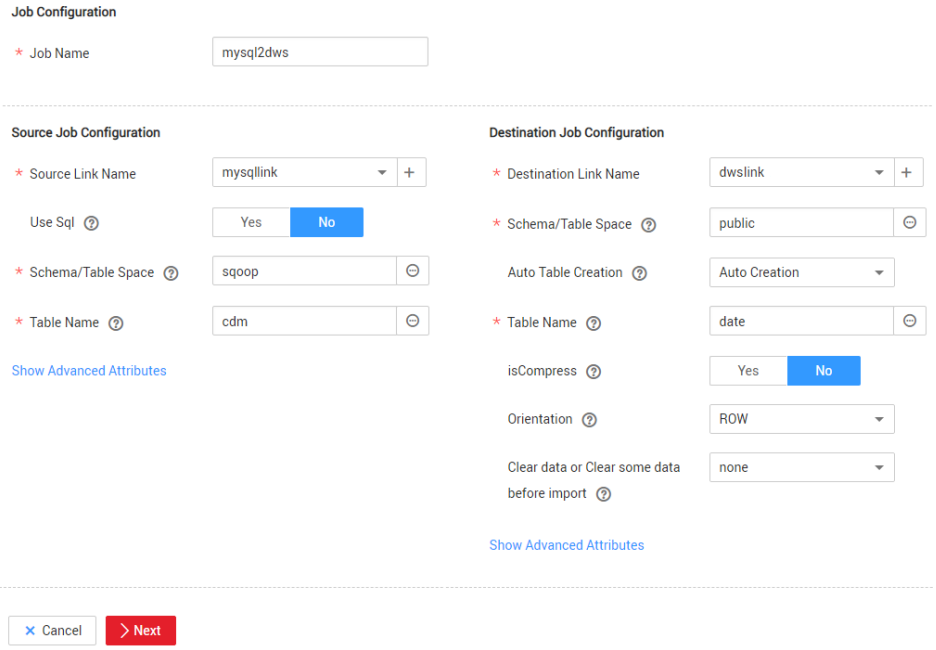
Procedure

Step 1 On the **Cluster Management** page, locate the **cdm-aff1** cluster created in **Step 1: Creating a Cluster**.

Step 2 Click **Job Management** in the **Operation** column of the CDM cluster.

Step 3 Choose **Table/File Migration** > **Create Job**, and configure the required job information.

Figure 2-5 Creating a job



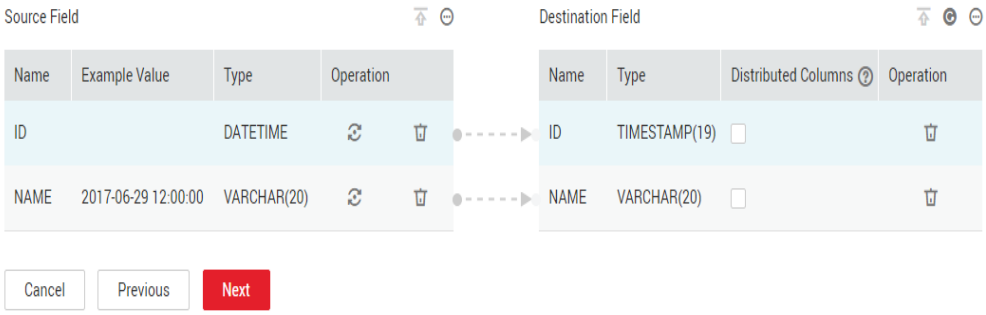
- **Job Name**: Enter a unique job name, for example, **mysql2dws**.
- **Source Job Configuration**
  - **Source Link Name**: Select the **mysqllink** link created in **Step 2: Creating Links**.
  - **Use SQL**: Select **No**.
  - **Schema/Tablespace**: Select the MySQL database from which the table is to be exported.
  - **Table Name**: Select the table from which data is to be exported.
  - Retain the default values of other optional parameters. For details, see **From a Relational Database**.
- **Destination Job Configuration**
  - **Destination Link Name**: Select the **dwslink** link created in **Step 2: Creating Links**.
  - **Schema/Tablespace**: Select the database to which data is to be imported.
  - **Auto Table Creation**: Select **Auto creation**. If the table specified by **Table Name** does not exist, CDM automatically creates the table in the DWS database.
  - **Table Name**: Select the table to which data is to be imported.
  - Retain the default values of other optional parameters. For details, see **To a Relational Database**.

Step 4 Click **Next**. The **Map Field** tab page is displayed. CDM automatically maps table fields at the migration source and destination. Check whether the field mapping is correct.

- If the field mapping is incorrect, click the row where the field is located and drag the field to adjust the mapping.

- When importing data to DWS, you need to manually select the distribution columns of DWS. 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 you need to convert the content of the source fields, perform the operations described in **Field Conversion**. In this example, the field conversion is not required.

Figure 2-6 Field mapping



Step 5 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**: To configure scheduled jobs, see **Scheduling Job Execution**. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
- **Write Dirty Data**: Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS for future viewing. Before writing dirty data, create an OBS link. Retain the default value **No** so that dirty data is not recorded.
- **Delete Job After Completion**: Retain the default value **Do not delete**.

Step 6 Click **Save and Run**. CDM starts to execute the job immediately.

NOTE

If the job fails to be executed, the following error message is displayed: SQL statements cannot be executed. ERROR: value too long for type character varying (7) Where: COPY dws_city, line 1, column name: '*Chinese characters*',

Cause: The length of the character field in the DWS table is insufficient. The encoding methods for Chinese characters stored in MySQL and DWS are different, and the required lengths are different as well. A Chinese character may occupy three bytes in UTF-8 encoding.

Solution: When creating a job in **Step 3**, enable automatic table creation. Set the **Extend Field Length** advanced attribute to **Yes**, and then execute the job again. In this way, when CDM automatically creates a table in DWS, the length of the character fields is set to three times that of the original table.

----End

2.5 Step 4: Viewing Job Execution Result

Scenario

This section describes how to view a job's execution results and its historical information in the latest 90 days, including the number of written rows, read rows, written bytes, written files, and log information.

Procedure

Step 1 On the **Cluster Management** page, locate the **cdm-aff1** cluster created in **Step 1: Creating a Cluster**.

Step 2 Click **Job Management** in the **Operation** column of the CDM cluster.

Step 3 Locate the **mysql_dws** job created in **Step 3: Creating and Executing a Job** and view the running status of the job.

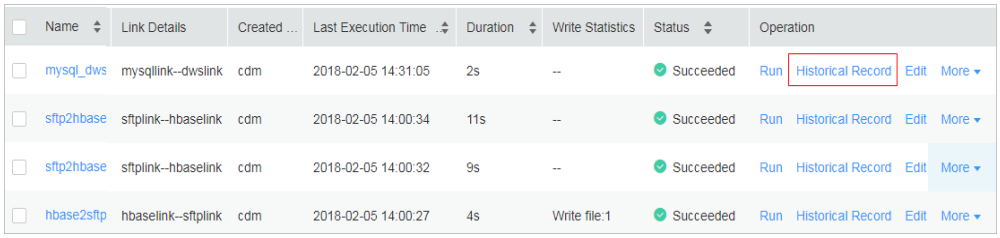
NOTE

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

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.

Step 4 Click **Historical Record** to view the number of written rows, number of read rows, number of written bytes, and number of written files.

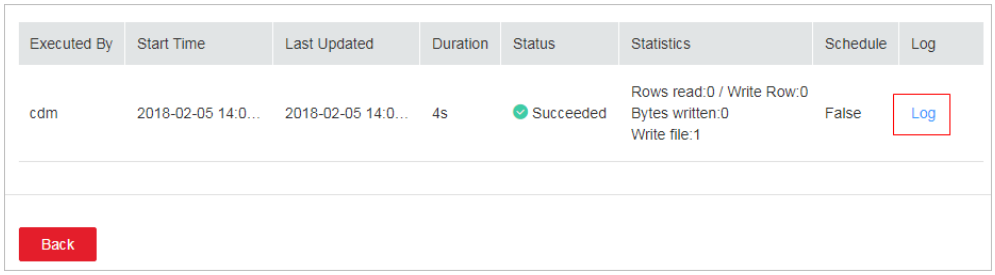
Figure 2-7 Viewing the historical records



Step 5 Click **Log** to view the job logs.

Alternatively, in the **Operation** column, choose **More** > **Log** to view the latest logs of the job.

Figure 2-8 Viewing job logs



----End

3 Cluster Management

3.1 Creating a CDM Cluster

Scenario

CDM provides isolated clusters to ensure secure and reliable data migration. Currently, each CDM cluster supports only one server. Subsequently, CDM clusters will support automatic capacity expansion.

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 VPC peering connection is configured, 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 public network for cross-VPC data migration, or contact the administrator to add specific routes to the VPC peering connection in the CDM background.

Procedure

Step 1 Log in to the **CDM management console**.

Step 2 Click **Buy CDM Cluster**. The page for buying a CDM cluster is displayed.

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

Table 3-1 Parameter description

Parameter	Example Value	Description
Region	CN North-Beijng1	Region where the CDM cluster resides. Resources in different regions cannot communicate with each other.

Parameter	Example Value	Description
AZ	AZ2	For details, see **Regions and AZs**.
Name	cdm-aff1	Custom CDM cluster name
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 20 jobs can be executed concurrently. This flavor is well suited to migrating a single database table with 10 million pieces of data or more. • **cdm.xlarge**: 16 vCPUs and 32 GB of memory The maximum and assured bandwidths are 10 Gbit/s and 4 Gbit/s. Up to 100 jobs can be executed concurrently. This flavor is well suited to TB-level data migration requiring 10GE high-speed bandwidth. • **cdm.4xlarge**: 64 vCPUs and 128 GB of memory The maximum and assured bandwidths are both 30 Gbit/s. Up to 300 jobs can be executed concurrently.
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	This parameter is available only when an enterprise project has been created on the **Enterprise Project Management** page. An enterprise project facilitates management of cloud resources. For more information, see the **Enterprise Management User Guide**.

Parameter	Example Value	Description
Auto Shutdown	No	After **Auto Shutdown** is enabled, if no job is running in the cluster and no scheduled job is created, a cluster will automatically shut down 15 minutes later to reduce costs. After a cluster is created, if you want to modify the configuration of auto shutdown or scheduled startup and shutdown, click the cluster name in the cluster list and click the **Cluster Configuration** tab. For details, see **Modifying Cluster Configurations**.
Scheduled Startup	No	The CDM cluster supports scheduled startup. If this parameter is enabled, set the scheduled startup time every day.
Scheduled Shutdown	No	During scheduled shutdown, the system does not wait for the completion of running jobs.
Notification	No	After the function is enabled, configure a maximum of five 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. **NOTE** The EIP exception notification takes effect only after the VPC policy agency of the corresponding region is created on the **IAM management console**. You can also choose **Authorize EIP Check** > **Create Agency** on the **Cluster Management** page to create an agency.

Step 4 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 5 Click **Submit**. The system starts to create a CDM cluster. You can view the creation progress on the **Cluster Management** page.

NOTE

Generally, it takes 10 to 20 minutes to create a cluster. If you create a CDM cluster for the first time, it takes only one minute to create it.

----End

3.2 Binding or Unbinding an EIP

Scenario

If you are binding an EIP or unbinding it from a CDM cluster, the EIPs you use are billed based on the VPC service billing rules.

- 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 a SNAT rule**.
- The EIP exception notification takes effect only after the VPC policy agency of the corresponding region is created on the **IAM management console**. You can also choose **Authorize EIP Check** > **Create Agency** on the **Cluster Management** page to create an agency.

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 management console**.

Step 2 In the left navigation pane, click **Cluster Management**. The **Cluster Management** page is displayed.

- Binding an EIP: In the **Operation** column, click **Bind EIP**. The **Bind EIP** dialog box is displayed.

Figure 3-1 Binding an EIP

- Unbinding an EIP: In the **Operation** column, choose **More** > **Unbind EIP**.

Step 3 Click **Yes**.

----End

3.3 Modifying Cluster Configurations

Configuration Description

After a CDM cluster is created, you can modify the following configurations:

- **Auto Shutdown**

  If no job is running in the cluster and no scheduled job is configured, the CDM cluster automatically shuts down after 15 minutes to avoid incurring any additional costs.

- **Scheduled Startup/Shutdown**

  Scheduled startup/shutdown and auto shutdown cannot be enabled at the same time. You can configure scheduled startup/shutdown at a specific time

every day. During scheduled shutdown, the system does not wait for unfinished jobs to complete.

- **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. Up to five mobile numbers and five email addresses can be configured.

- **User Isolation**
  - Enabled

    Migration jobs and links in the cluster are isolated. Other IAM users under the same cloud account cannot view the jobs and links.
  - Disabled

    Migration jobs and links in the cluster can be shared by users. All IAM users with CDM Administrator permissions under the same cloud account can view and perform operations on the jobs and links.After disabling **User Isolation**, restart the cluster VM for the settings to take effect.

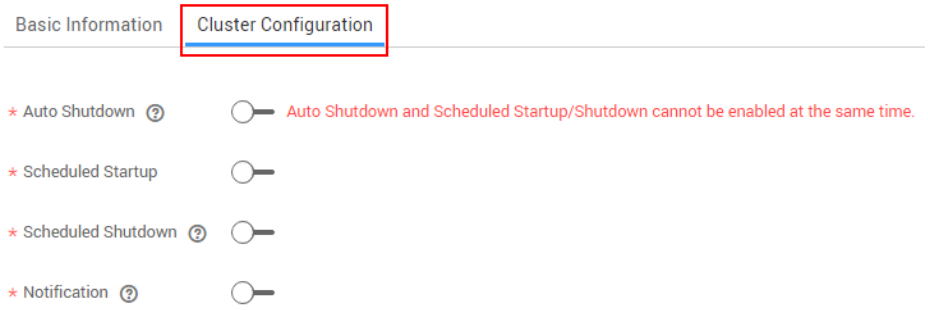
Procedure

Step 1 Log in to the **CDM management console**.

Step 2 In the left navigation pane, click **Cluster Management**. The **Cluster Management** page is displayed.

Step 3 Click the name of a cluster and click the **Cluster Configuration** tab to modify the configuration of auto shutdown, scheduled startup/shutdown, notification, or user isolation.

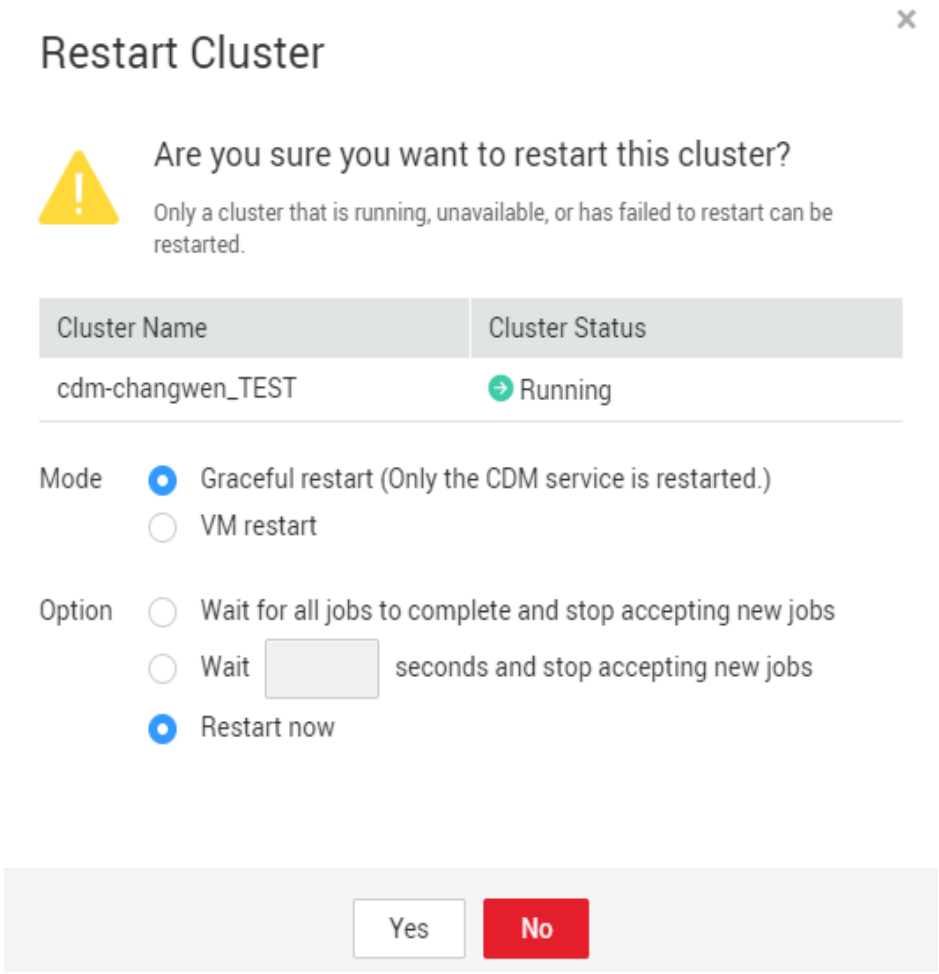
Figure 3-2 Modifying cluster configurations



Step 4 Click **Save**. The **Cluster Management** page is displayed.

Step 5 If **User Isolation** is disabled, choose **More** > **Restart** in the **Operation** column to restart the cluster VM for the settings to take effect.

Figure 3-3 Restarting a cluster



- **Graceful restart**: 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 6 Select **VM restart** and click **Yes**.

----End

3.4 Viewing Cluster Configurations, Logs, and Monitoring Data

Scenario

This section describes how to view cluster configurations, obtain cluster logs, and view monitoring data on Cloud Eye.

Prerequisites

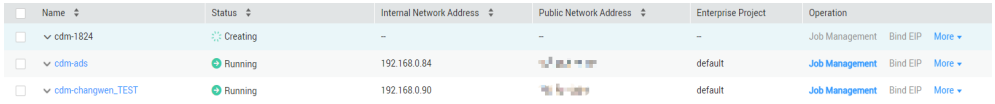
You have created a CDM cluster.

Procedure

Step 1 Log in to the **CDM management console**.

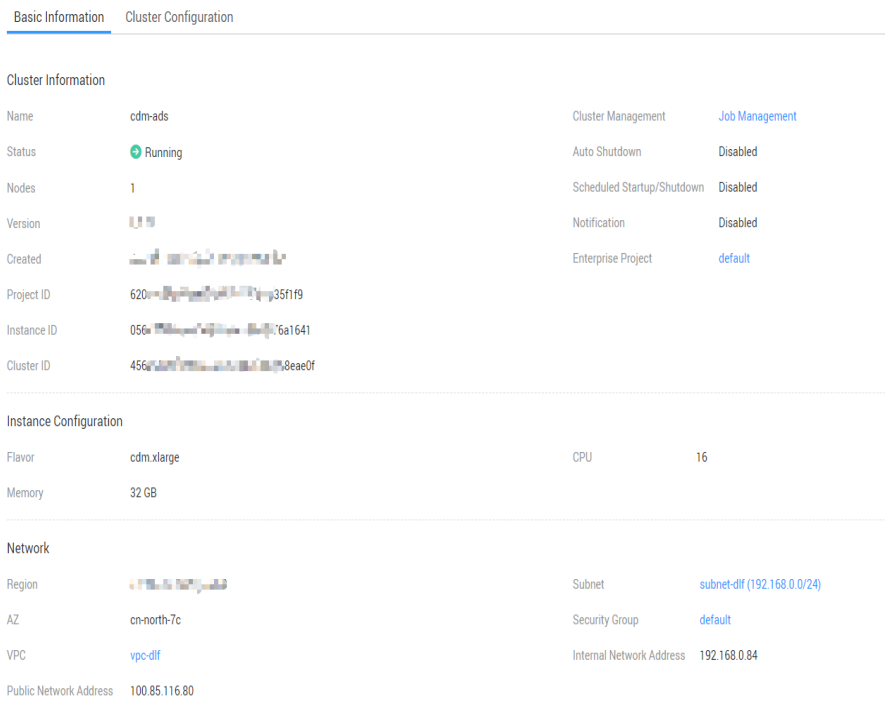
Step 2 In the left navigation pane, click **Cluster Management** to display the cluster list.

Figure 3-4 Cluster list



Step 3 Click the cluster name to view basic details of the cluster, including the cluster flavor, creation time, node quantity, node configurations, network configurations, project ID, cluster ID, and instance ID.

Figure 3-5 Cluster configurations



Step 4 In the row of the cluster, choose **More** > **Download Log** to obtain cluster logs.

Step 5 In the **Operation** column, choose **More** > **View Metric**. The Cloud Eye management console is displayed, and you can see the inbound and outbound rates, and CPU and memory usages. For details about the monitoring metrics, see **CDM Metrics**.

----End

3.5 Monitoring

3.5.1 CDM Metrics

Function

This section describes metrics reported by CDM to Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query metric information generated for CDM.

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

Dimension

Key	Value
instance_id	CDM instance

3.5.2 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 Log in to the **CDM management console**.

Step 2 Choose **Cluster Management**. Choose **More** > **View Metric**. The Cloud Eye console is displayed.

Step 3 In the left navigation pane of the Cloud Eye console, choose **Alarm Management** > **Alarm Rules** > **Create Alarm Rule**.

Step 4 Set the alarm rule for the CDM cluster as prompted.

Step 5 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 CDM alarm rules, see **Cloud Eye User Guide**.

----End

3.5.3 Querying Metrics

Scenario

Cloud Eye monitors the running status of the CDM cluster. You can obtain the monitoring metrics of CDM on the Cloud Eye management console.

Monitored data requires a period of time for transmission and display. The status of CDM displayed on the Cloud Eye page 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 shut down or restart, or is unavailable, its monitoring metrics cannot be viewed on Cloud Eye. 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.

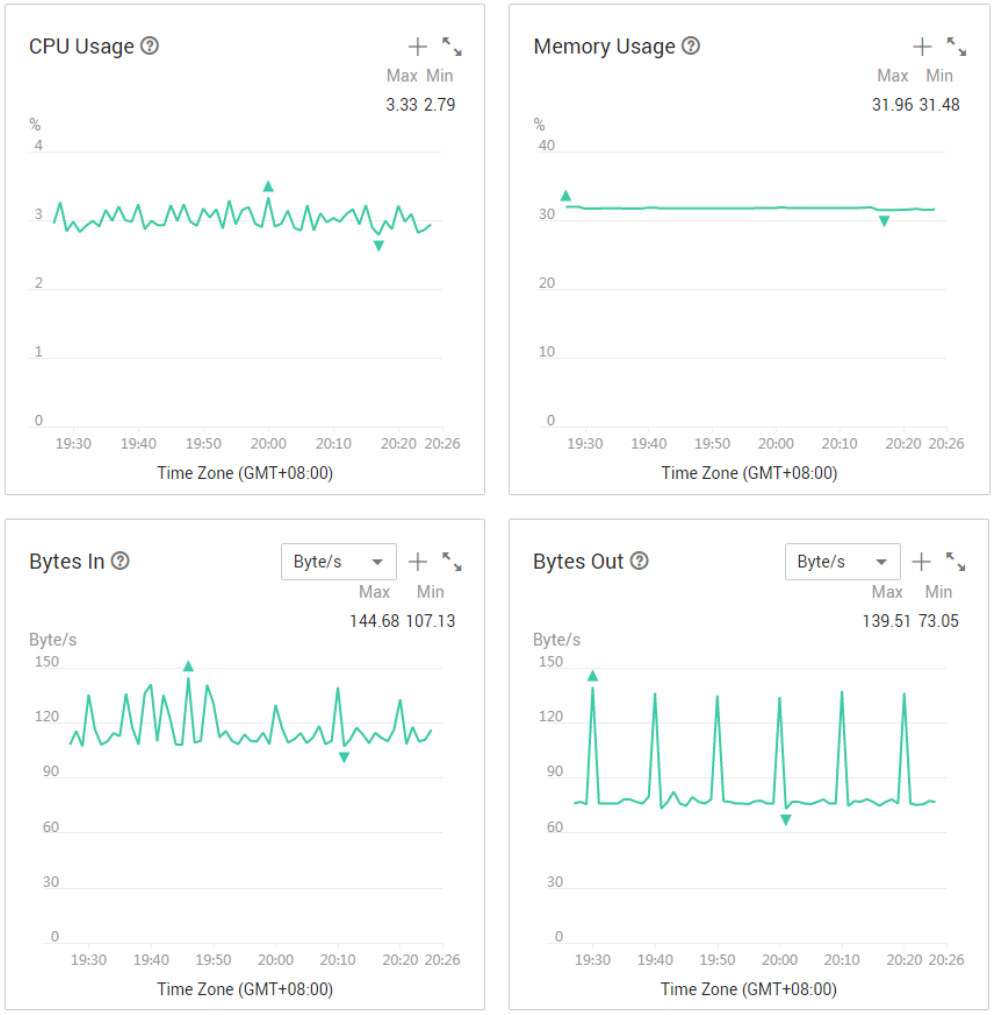
Procedure

Step 1 Log in to the **CDM management console**.

Step 2 Choose **Cluster Management**. Choose **More** > **View Metric**. The Cloud Eye console is displayed.

Step 3 On the CDM monitoring page, you can view the graphs of all monitoring metrics.

Figure 3-6 Viewing metrics

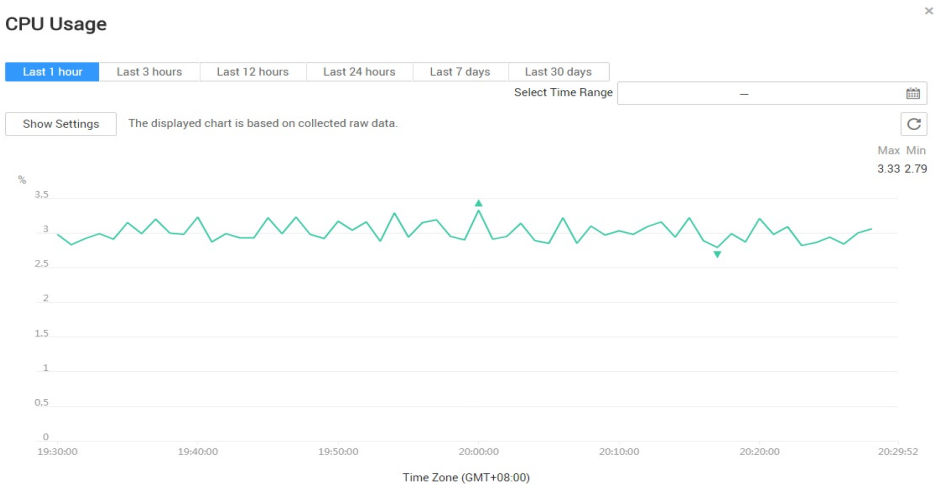


Step 4 Click in the upper right corner of the graphs to zoom in the graphs.

The system allows you to select a fixed time range or customize the time range.

1. Fixed time ranges include **Last 1 hour**, **Last 3 hours**, **Last 12 hours**, **Last 24 hours**, **Last 7 days**, and **Last 30 days**.
2. A customized time range can be specified within the latest seven days.

Figure 3-7 Zoomed in monitoring graph



----End

3.6 CTS

3.6.1 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 3-3 CDM 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
Stopping a cluster	cluster	stopCluster
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
Deleting a link	link	deleteLink

Operation	Resource Type	Trace Name
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

3.6.2 Viewing Traces

Scenario

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.

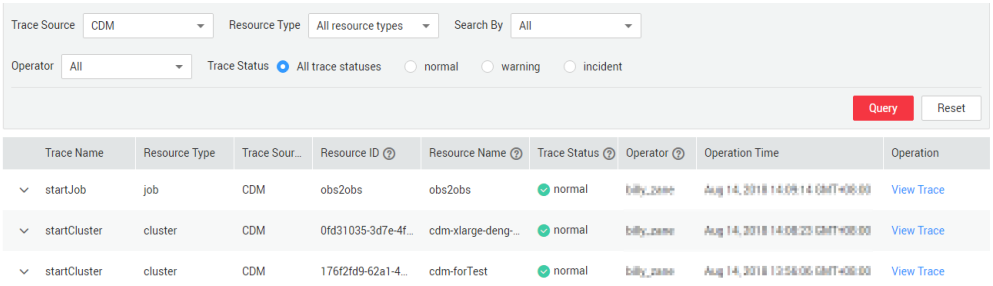
This section describes how to query these traces.

Procedure

1. Log in to the management console.
2. Click **Service List**, and choose **Management & Deployment** > **Cloud Trace Service**.
3. In the left navigation pane, click **Trace List**.

   Click **Filter** and specify filter criteria as needed.

   Figure 3-8 CDM traces



4. Unfold the target trace to view its details.
5. Click **View Trace** in the **Operation** column to view the trace structure details.

   For more information about CTS, see **Cloud Trace Service User Guide**.

4 Link Management

4.1 Creating Links

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 or scenario-based migration), see **Supported Data Sources**.

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

Prerequisites

- You have created a cluster as described in **Creating a CDM Cluster**.
- The CDM cluster can communicate with the data source. To connect the intranet to the cloud service, perform the operations in **How Do I Connect the On-Premises Intranet or Third-Party Private Network to CDM**.
- 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.

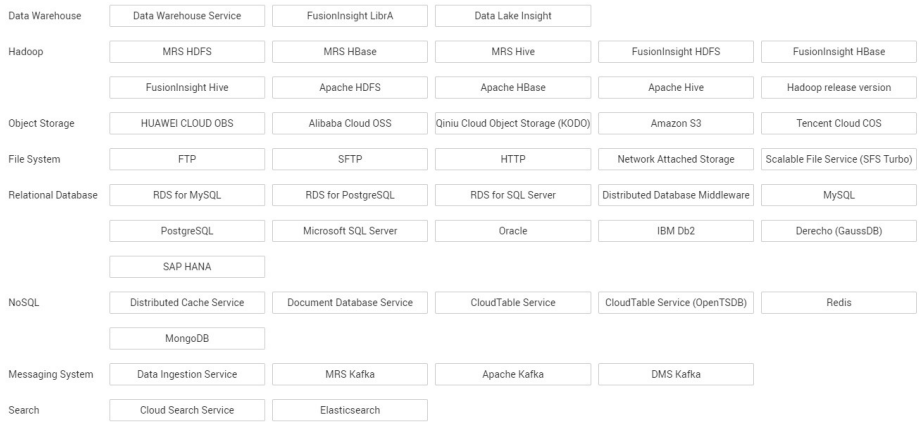
Procedure

Step 1 Log in to the **CDM management console**.

Step 2 In the left navigation pane, click **Cluster Management**. Locate the target cluster, choose **Job Management** > **Link Management** > **Create Link**, and select a connector. See **Figure 4-1**.

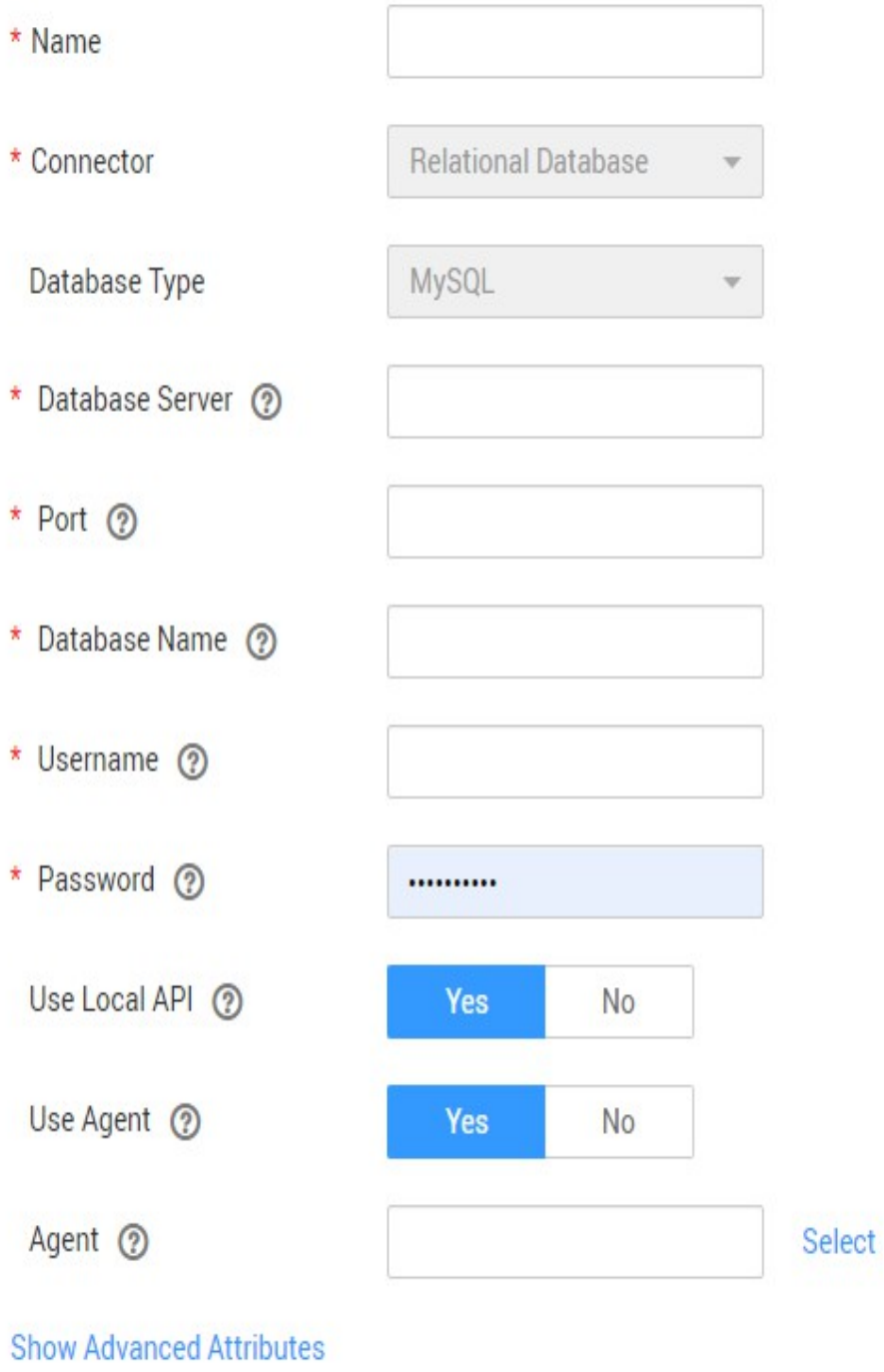
The connectors are classified based on the type of the data source to be connected. All supported data types are displayed.

Figure 4-1 Selecting a connector



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

Figure 4-2 Creating 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
• Data Warehouse Service • RDS for MySQL • RDS for PostgreSQL • RDS for SQL Server • MySQL • PostgreSQL • Microsoft SQL Server • Oracle • IBM Db2 • FusionInsight LibrA • Derecho (GaussDB) • NewSQL (GaussDB) • SAP HANA • MYCAT • Dameng database • Sharding	Because the JDBC drivers used to connect to these relational databases are the same, the parameters to be configured are also the same and are described in **Link to Relational Databases**. • When importing data to DWS, specify the **COPY** or **GDS** import mode to improve the import performance. You can specify the **Import Mode** parameter when **creating a DWS link**. • When importing data to RDS for MySQL, enable the LOAD DATA function of MySQL to accelerate data import and improve the import performance. You can configure **Use Local API** to enable the function when you **create a MySQL link**.
HUAWEI CLOUD OBS	If the data source is OBS, see **Link to OBS**.
Alibaba Cloud OSS	If the data source is OSS on Alibaba Cloud, see **Link to OSS on Alibaba Cloud**. Currently, data can only be exported from OSS to OBS.
Qiniu Cloud Object Storage (KODO) Tencent Cloud COS	If the data source is KODO or COS, see **Link to KODO/COS**. Currently, data can only be exported from KODO/COS to OBS.
Amazon S3	If the data source is Amazon S3, see **Link to Amazon S3**. Currently, objects can only be exported from Amazon S3 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**. **NOTE** If **Run Mode** is set to **Standalone**, CDM can migrate data between HDFSs of multiple MRS clusters.
• MRS HBase • FusionInsight HBase • Apache HBase	If the data source is HBase of MRS, Apache Hadoop, or FusionInsight HD, see **Link to HBase**.

Connector	Description
• MRS Hive • FusionInsight Hive • Apache Hive	If the data source is Hive of MRS, 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 • HTTPS	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.
• NAS • SFS Turbo	If the data source is a NAS server, see **Link to NAS/SFS**. CIFS, SMB, and NFS are supported. CDM can connect to dedicated file servers, Windows file sharing servers, Linux Samba servers, and cloud services that support CIFS, SMB, or NFS file systems such as SFS.
• MongoDB • Document Database Service	If the data source is a local MongoDB or DDS, see **Link to MongoDB**.
• Redis • Distributed Cache Service	If the data source is a local Redis database or DCS, see **Link to Redis/DCS**. Currently, data can be imported to but cannot be exported from DCS. Data can be imported to and exported from the open source Redis.
Apache Kafka	If the data source is the open source Kafka, see **Link to Kafka**. Currently, data can only be exported from Kafka to CSS, DIS, or DMS Kafka.
Data Ingestion Service	If the data source is DIS, see **Link to DIS**. Currently, data can only be exported from DIS to CSS, Apache Kafka, or DMS Kafka.
• Cloud Search Service • Elasticsearch	If the data source is CSS or Elasticsearch, see **Link to Elasticsearch/CSS**.

Connector	Description
Data Lake Insight	If the data source is DLI, see **Link to DLI**. Currently, data can be imported to but cannot be exported from DLI.
OpenTSDB	If the data source is OpenTSDB, see **Link to CloudTable OpenTSDB**.
DMS Kafka	If the data source is DMS Kafka, see **Link to DMS Kafka**. Currently, data can only be exported from DMS Kafka to CSS, Apache Kafka, DIS, or DMS Kafka.

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

4.2 Managing Drivers

The Java Database Connectivity (JDBC) API provides programmatic access to relational databases. Using the JDBC API, applications can execute SQL statements and retrieve results. Before connecting CDM to a relational database, you need to upload a driver.

Prerequisites

- You have created a cluster as described in **Creating a CDM Cluster**.
- 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?

Different types of relational databases need to adapt to different drivers. Download one of the drivers listed in **Table 4-2** based on the preset driver name on the **Driver Management** page.

Table 4-2 Drivers

Relational Database Type	Driver Name	How to Obtain	Recommended Version
• RDS for MySQL • MySQL	MYSQL MYCAT	https://downloads.mysql.com/archives/c-j/	5.1.48. The driver of the 8.x version is not supported.
Oracle	ORACLE_6 ORACLE_7 ORACLE_8	https://www.oracle.com/database/technologies/appdev/jdbc-downloads.html	Obtain **ojdbc8.jar** in **Oracle Database 12c Release 2 (12.2.0.1) drivers**.
• RDS for PostgreSQL • PostgreSQL	POSTGRESQL	https://jdbc.postgresql.org/download.html	42.1.4
IBM Db2	DB2	https://www.ibm.com/support/pages/db2-jdbc-driver-versions-and-downloads	4.21.29
• RDS for SQL Server • Microsoft SQL Server	SQLServer	https://docs.microsoft.com/en-us/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?view=sql-server-ver15	4.2

NOTE

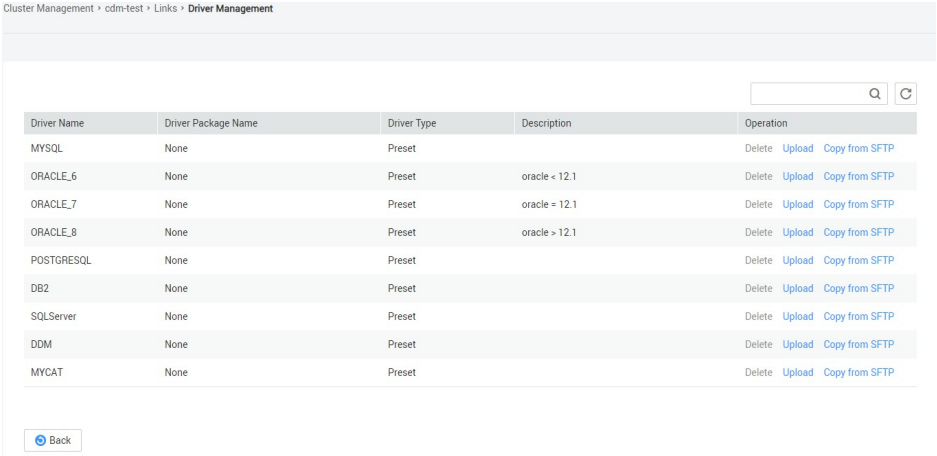
Drivers for GaussDB and SAP HANA databases have been loaded to the system. You do not need to upload the drivers again.

Procedure

Step 1 Log in to the **CDM management console**.

Step 2 In the navigation pane, choose **Cluster Management**. Locate the target cluster and choose **Job Management** > **Link Management** > **Driver Management**. The **Driver Management** page is displayed.

Figure 4-3 Uploading a driver



Step 3 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.

----End

4.3 Link to Hive

CDM supports the following Hive data sources:

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

MRS Hive

The MRS Hive link is used for MapReduce Service (MRS) on HUAWEI CLOUD. **Table 4-3** describes related parameters.

NOTE

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.

Table 4-3 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 if MRS is in non-security mode. • **KERBEROS**: Select this if MRS is in 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.	cdm
Password	Password used for logging in to MRS Manager	-
OBS storage support	The server must support OBS storage. When creating a Hive table, you can store the table in OBS.	No
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, 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.	EMBEDDED

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

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 4-4 describes related parameters.

Table 4-4 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	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
Manager Port	FusionInsight/MRS Manager port	28443
CAS Server Port	CAS protocol port of FusionInsight/MRS Manager	20009
Authentication Method	Authentication method used for accessing MRS • **SIMPLE**: Select this if MRS is in non-security mode. • **KERBEROS**: Select this if MRS is in security mode.	SIMPLE
HIVE Version	Hive version	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.	cdm
Password	Password used for logging in to MRS 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
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, 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.	EMBEDDED

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

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 4-5 describes related parameters.

Table 4-5 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

Parameter	Description	Example Value
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 MRS • **SIMPLE**: Select this if MRS is in non-security mode. • **KERBEROS**: Select this if MRS is in 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
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, 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.	EMBEDDED

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

4.4 Link to CloudTable

When connecting CDM to CloudTable, configure the parameters as described in **Table 4-6**.

Table 4-6 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	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
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. If you have not created access keys for your account, create them as described in **(Optional) Obtaining the Authentication Information**.	-
SK	SK for accessing the CloudTable cluster. For details about how to obtain the SK, see the description for **AK**.	-

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, 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.	EMBEDDED

4.5 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 OBS or a database.

When connecting CDM to an FTP or SFTP server, configure the parameters as described in **Table 4-7**.

Table 4-7 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	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, which is 21 by default	21

Parameter	Description	Example Value
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	-

4.6 Link to NAS/SFS

When connecting CDM to a NAS server or SFS Turbo, configure the parameters as described in **Table 4-8**.

CIFS, SMB, and NFS are supported. CDM can connect to dedicated file servers, Windows file sharing servers, Linux Samba servers, and cloud services that support CIFS, SMB, or NFS file systems such as SFS.

Table 4-8 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	nas_link
Protocol	NAS file protocol. Currently, only SMB, CIFS, and NFS protocols are supported.	NFS
Shared Path	Shared path of the NAS server	\\server\share
Username	Username for logging in to the NAS server, which is in the *domain name*|*username* format This parameter is not displayed if **Protocol** is set to **NFS**.	domain01\user
Password	Password for logging in to the NAS server This parameter is not displayed if **Protocol** is set to **NFS**.	-

4.7 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 4-9**.

Table 4-9 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
MongoDB 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 MongoDB database to be connected	DB_mongodb
Username	Username for logging in to MongoDB	cdm
Password	Password for logging in to MongoDB	-

4.8 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 4-10**.

Table 4-10 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	-

4.9 Link to Redis/DCS

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.

The DCS link is used to load data from databases or files to Distributed Cache Service (DCS) on HUAWEI CLOUD. You are advised to use backup and restoration to migrate data from the third-party cloud Redis services to DCS.

When connecting CDM to an on-premises Redis database or DCS, configure the parameters as described in **Table 4-11**.

Table 4-11 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	redis_link
Redis Deployment Method	Two deployment methods are available: • **Single**: installation on a single-node system • **Cluster**: installation on a cluster	Single
Redis 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
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

4.10 Link to Kafka

When connecting CDM to MRS, FusionInsight, or Kafka of local Apache Hadoop, configure the parameters as described in **Table 4-12**.

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.

Table 4-12 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
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.	-
Authentication Method	Authentication method used for accessing MRS • **SIMPLE**: for non-security mode • **KERBEROS**: for security mode	Yes
Username	Username used for logging in to MRS Manager	-
Password	Password used for logging in to MRS Manager	-

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

4.11 Link to DIS

When connecting CDM to DIS, configure the parameters as described in **Table 4-13**. Currently, data can only be exported from DIS to CSS, Apache Kafka, or DMS Kafka.

Table 4-13 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	dis_link
Region	Region where DIS is deployed	cn-north-1
Endpoint	URL of DIS in the format of https://*Endpoint*. Obtain the endpoint from **Regions and Endpoints**.	https://dis.cn-north-1.myhuaweicloud.com

Parameter	Description	Example Value
AK	AK used for logging in to the DIS server. If you have not created access keys for your account, create them as described in **(Optional) Obtaining the Authentication Information**.	-
SK	SK used for logging in to the DIS server. For details about how to obtain the SK, see the description for **AK**.	-
Project ID	Project ID of DIS. For details about how to obtain a project ID, see **(Optional) Obtaining the Authentication Information**.	-

4.12 Link to Elasticsearch/CSS

Elasticsearch

The Elasticsearch link is applicable to data migration of third-party cloud Elasticsearch services and Elasticsearch created in the local data center or ECS.

When connecting CDM to Elasticsearch, configure the parameters as described in **Table 4-14**.

Table 4-14 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	css_link
Elasticsearch Server List	List of one or more Elasticsearch servers, including the port number. The format is *ip:port*. Use semicolons (;) to separate multiple IP addresses.	192.168.0.1:9200; 192.168.0.2:9200

CSS

The Cloud Search Service (CSS) link is used to migrate log files or database records to the Elasticsearch engine for search and analysis.

When connecting CDM to CSS, configure the parameters as described in **Table 4-15**.

Table 4-15 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	css_link
Elasticsearch Server List	List of one or more Elasticsearch servers, including the port number. 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.	-

4.13 Link to DLI

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

Table 4-16 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	dli_link
AK	AK required for authentication during access to the DLI database. If you have not created access keys for your account, create them as described in **(Optional) Obtaining the Authentication Information**.	-
SK	SK required for authentication during access to the DLI database. For details about how to obtain the SK, see the description for **AK**.	-

Parameter	Description	Example Value
Project ID	Project ID in the region where DLI resides	-

4.14 Link to CloudTable OpenTSDB

When connecting CDM to CloudTable OpenTSDB, configure the parameters as described in **Table 4-17**.

Table 4-17 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	TSDB_link
OpenTSDB Link	ZK link of OpenTSDB	opentsdb-sp8afz7bgbps5ur.cloudtable.com:4242
Security Mode	Security or non-security mode If you select **Security**, enter the project ID, username, and AK/SK.	Nonsecurity
Project ID	Project ID in the region where CloudTable resides	-
Username	Username for accessing CloudTable	admin
AK	AK for accessing CloudTable. If you have not created access keys, create them as described in **Obtaining Authentication Information**.	-
SK	SK for accessing CloudTable. For details about how to obtain the SK, see the description for **AK**.	-

4.15 Link to DMS Kafka

When connecting CDM to DMS Kafka, configure the parameters as described in **Table 4-18**.

Table 4-18 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	dms_link
Kafka Broker	Address of a Kafka premium instance. The format is host:port.	-
Kafka SASL_SSL	Whether to enable SSL authentication when a client connects to a Kafka premium instance. If Kafka SASL_SSL is enabled, data will be encrypted before transmission, providing higher security.	Yes
Username	Username used for connecting to the Kafka premium instance	-
Password	Password used for connecting to the Kafka premium instance	-

4.16 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 4-19**.

Table 4-19 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.	cdm
Password	Password used for logging in to MRS Manager	-
Authentication Method	Authentication method used for accessing MRS • **SIMPLE**: Select this if MRS is in non-security mode. • **KERBEROS**: Select this if MRS is in security mode.	SIMPLE
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, 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

FusionInsight HBase

When connecting CDM to HBase of FusionInsight HD, configure the parameters as described in **Table 4-20**.

Table 4-20 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 FusionInsight HD • **SIMPLE**: Select this if FusionInsight HD is in non-security mode. • **KERBEROS**: Select this if FusionInsight HD is in security mode.	Kerberos
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, 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

Apache HBase

When connecting CDM to HBase of Apache Hadoop, configure the parameters as described in **Table 4-21**.

Table 4-21 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 Hadoop • **SIMPLE**: Select this if Hadoop is in non-security mode. • **KERBEROS**: Select this if Hadoop is in security mode. Obtain the **Principal** account and **Keytab File** file of the client for authentication.	Kerberos
Principal	When **Authentication Method** is set to **KERBEROS**, the **Principal** account is used for authentication. You can contact the Hadoop administrator to obtain the account.	USER@YOUR-REALM.COM
Keytab File	When **Authentication Method** is set to **KERBEROS**, this file is used for authentication. You can contact the Hadoop administrator to obtain the file.	/opt/user.keytab
IP and Host Name Mapping	If the 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.	10.3.6.9 hostname01 10.4.7.9 hostname02

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, 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

4.17 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 4-22**.

Table 4-22 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

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
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.	cdm
Password	Password used for logging in to MRS Manager	-
Authentication Method	Authentication method used for accessing MRS • **SIMPLE**: Select this if MRS is in non-security mode. • **KERBEROS**: Select this if MRS is in 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 **STANDALONE** is selected, CDM can migrate data between HDFSs of multiple MRS clusters.	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**.	-

FusionInsight HDFS

When connecting CDM to HDFS of FusionInsight HD, configure the parameters as described in **Table 4-23**.

Table 4-23 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

Parameter	Description	Example Value
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 FusionInsight HD • **SIMPLE**: Select this if FusionInsight HD is in non-security mode. • **KERBEROS**: Select this if FusionInsight HD is in 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.	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**.	-

Apache HDFS

When connecting CDM to HDFS of Apache Hadoop, configure the parameters as described in **Table 4-24**.

Table 4-24 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	hdfs://nn1.example.com/
Authentication Method	Authentication method used for accessing Hadoop • **SIMPLE**: Select this if Hadoop is in non-security mode. • **KERBEROS**: Select this if Hadoop is in security mode. Obtain the **Principal** account and **Keytab File** file of the client for authentication.	KERBEROS
Principal	When **Authentication Method** is set to **KERBEROS**, the **Principal** account is used for authentication. You can contact the Hadoop administrator to obtain the account.	USER@YOUR-REALM.COM
Keytab File	When **Authentication Method** is set to **KERBEROS**, this file is used for authentication. You can contact the Hadoop administrator to obtain the file.	/opt/user.keytab

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
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
Agent	If **Run Mode** is set to **Agent**, click **Select** and select the agent created in **Connecting to an Agent**.	-

4.18 Link to Amazon S3

When connecting CDM to Amazon S3, configure the parameters as described in **Table 4-25**. Currently, objects can only be exported from Amazon S3 to OBS.

Table 4-25 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	linkname
Endpoint	Endpoint of the Amazon S3 bucket	-
Region	Region to which the Amazon S3 bucket belongs	-
AK	AK for accessing the bucket of Amazon S3	-
SK	SK for accessing the bucket of Amazon S3	-

4.19 Link to KODO/COS

When connecting CDM to KODO or COS, configure the parameters as described in **Table 4-26**. Currently, data can only be exported from KODO/COS to OBS.

Table 4-26 KODO/COS 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	linkname
Region	Region where KODO/COS resides	z0
AK	AK for accessing KODO/COS	-
SK	SK for accessing KODO/COS	-
Use Custom Domain Name to Download Objects	This parameter is available for KODO links. Whether to preferentially use the custom domain name to download objects from a bucket if the object storage bucket hosts a CDN acceleration domain name or other custom domain names.	Yes

4.20 Link to OSS on Alibaba Cloud

When connecting CDM to OSS on Alibaba Cloud, 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	oss_link
OSS Endpoint	Endpoint of Alibaba Cloud OSS	oss-cn-hangzhou.aliyuncs.com
Authentication Method	Available identity authentication methods: • **Access Key**: Use the access key to access OSS. • **Security Token Service**: Use the temporary key and security token to access OSS.	Access Key
AK	AK used for logging in to the OSS server	-
SK	SK used for logging in to the OSS server	-
Security Token	Enter the temporary token provided by Security Token Service (STS).	-
IP and domain Name Mapping	Mapping between IP addresses and domain names	-

4.21 Link to Relational Databases

CDM supports the following relational databases:

- Data Warehouse Service
- RDS for MySQL
- RDS for PostgreSQL
- RDS for SQL Server
- MySQL
- PostgreSQL
- Microsoft SQL Server
- Oracle
- IBM Db2
- FusionInsight LibrA
- Derecho (GaussDB)
- NewSQL (GaussDB)
- SAP HANA

- MYCAT
- Dameng database
- Sharding

Prerequisites

You have uploaded the required driver by following the instructions provided in **Managing Drivers**.

Link Parameters

Table 4-28 describes the required parameters of the link to DWS, RDS for MySQL, RDS for PostgreSQL, RDS for SQL Server, MySQL, PostgreSQL, Microsoft SQL Server, Oracle, IBM Db2, or Derecho (GaussDB).

Table 4-28 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	mysql_link
Database Server	IP address or domain name of the database to be connected Click **Select** next to the text box to obtain the list of DWS and RDS instances.	192.168.0.1
Port	Port number of the database to be connected	3306
Connection Type	This parameter is available only for Oracle database links. The options are as follows: • **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 an Oracle link is created and **Database Link Type** is set to **SID**.	dbname
Database Name	Name of the database to be connected	dbname
Username	Username used for accessing the database. This account must be able to read and write data tables and read metadata of the database.	cdm
Password	Password of the account	-
Use Agent	Whether to extract data from the data source through an agent	Yes

Parameter	Description	Example Value
Agent	Click **Select** and select the agent created in **Connecting to an 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
Fetch Size	(Optional) This parameter is displayed only after 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
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 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
SSL Encryption	(Optional) If you set this parameter to **Yes**, CDM can connect to RDS (on-premises databases excluded) in SSL encryption mode. Security hardening has been performed on RDS for PostgreSQL. For this reason, when creating a link to RDS for PostgreSQL, set this parameter to **Yes**.	Yes
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

Parameter	Description	Example Value
Reference Sign	(Optional) Delimiter between the names of the referenced tables or columns. For details, see the product documentation of the corresponding database.	'
Oracle Version	This parameter is displayed only when you create an Oracle link. Select an option according to the version of the Oracle database. If the error message, "java.sql.SQLException: Protocol violation", is displayed, select another option.	Later than 12.1

4.22 Link to OBS

When connecting CDM to OBS, configure the parameters as described in **Table 4-29**.

Table 4-29 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
OBS Endpoint	Endpoint of the OBS bucket, which can be obtained from **Regions and Endpoints** You can enter a bucket-level domain name, for example, **test.obs.myhuaweicloud.com**. In this case, you can query only the **test** bucket.	obs.cn-north-1.myhuaweicloud.com
Port	Data transmission port. The HTTPS port number is 443 and the HTTP port number is 80.	443
AK	AK used for logging in to the OBS server. If you have not created access keys for your account, create them as described in **Obtaining Authentication Information**.	-
SK	SK used for logging in to the OBS server. SK is used together with AK. For details about how to obtain the SK, see the description for **AK**.	-

4.23 Editing/Deleting a Link

Scenario

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

- Edit: You can modify link parameters, but cannot re-select connectors. To modify a link, you need to re-enter the password needed to access the data source.
- Test Connectivity: You can test the connectivity of a created link directly.
- View Link JSON: Check the link parameter settings in JSON format.
- Edit Link JSON: Edit the link parameter settings in JSON format.
- Delete: You can delete links that are not used by any jobs in batches.

Prerequisites

- You have obtained the username and password needed to access the desired data source.
- Links are not used by any jobs.

Procedure

Step 1 Log in to the **CDM management console**.

Step 2 In the left navigation pane, click **Cluster Management**. Locate the target cluster and choose **Job Management** > **Link Management**.

Step 3 On the **Link Management** page, locate the links to be deleted.

- Edit: 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.
- Test Connectivity: Click **Test Connectivity** in the **Operation** column to test the connectivity of the created link.
- View Link JSON: In the **Operation** column, choose **More** > **View Link JSON** to view link parameters in JSON format.
- Edit Link JSON: In the **Operation** column, choose **More** > **Edit Link JSON** to modify link parameters in JSON format.
- Delete: Select multiple links, and click **Delete Link** next to **Create Link** to batch delete unused links.

----End

5 Job Management

5.1 Table/File Migration Jobs

Scenario

CDM can migrate tables or files between homogeneous and heterogeneous data sources. For details about data sources that support table/file migration, see **Data Sources Supported by CDM**.

CDM is applicable to data migration to the cloud, data exchange on the cloud, and data migration to on-premises service systems.

Prerequisites

- You have created links based on the instructions in **Creating Links**.
- The CDM cluster can communicate with the data source.

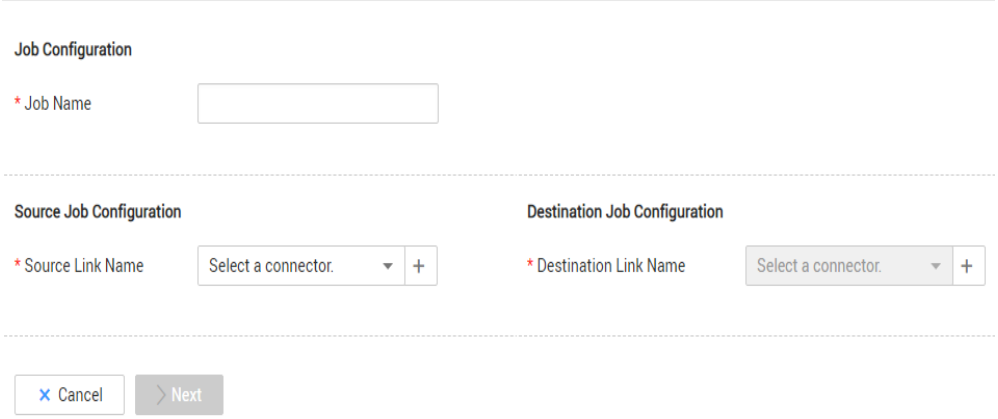
Procedure

Step 1 Log in to the **CDM management console**.

Step 2 In the left navigation pane, click **Cluster Management**. Locate the target cluster and click **Job Management**.

Step 3 Choose **Table/File Migration** > **Create Job**. The page for configuring the job is displayed.

Figure 5-1 Creating a migration job



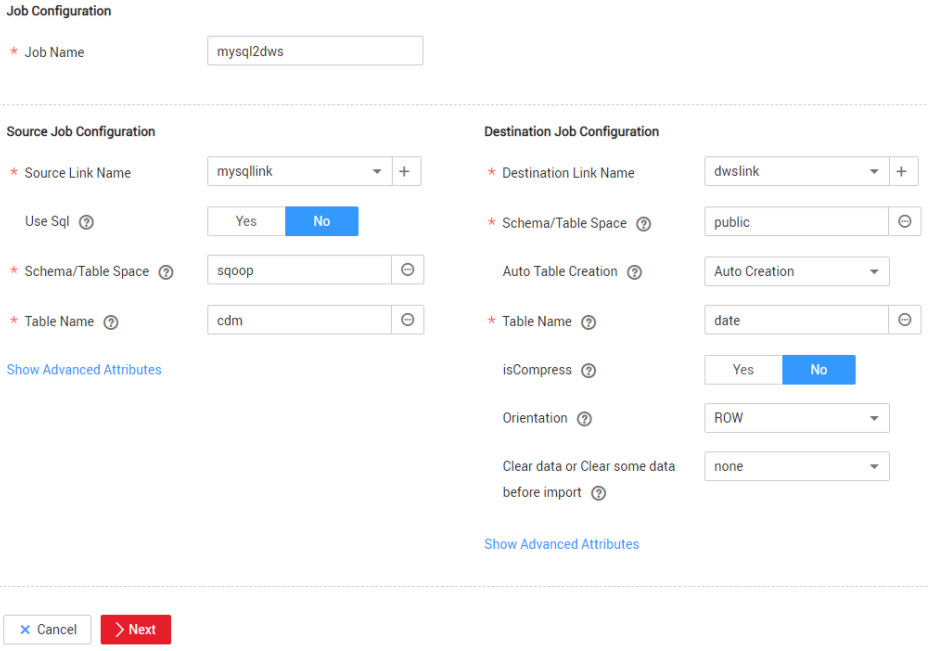
Step 4 Select the source and destination links.

- **Job Name**: Enter a custom job name, which is a string of 1 to 256 characters chosen from letters, underscores (_), and digits, for example, **oracle2obs_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.

If no link is available, click **+** or go to the **Link Management** page to create one. For details about how to create a link, see **Creating Links**.

Step 5 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 • Alibaba Cloud OSS • KODO • COS	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. Currently, data cannot be imported to Alibaba Cloud OSS, KODO, and COS.	For details, see **From OBS/OSS/KODO/COS/S3**.
• 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	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**.
• FTP • SFTP • Network Attached Storage • SFS Turbo	FTP, SFTP, NAS, or SFS data can be exported in CSV, JSON, or binary format.	For details, see **From FTP/ SFTP/NAS/SFS**.

Migration Source	Description	Parameter Settings
• HTTP • HTTPS	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/HTTPS URLs.	For details, see **From HTTP/HTTPS**.
• Data Warehouse Service • RDS for MySQL • RDS for SQL Server • RDS for PostgreSQL • Dameng database	Data can be exported from the cloud database services.	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 a Relational Database**.
• FusionInsight LibrA • Derecho (GaussDB)	Data can be exported from FusionInsight LibrA and Derecho.	
• MySQL • PostgreSQL • Oracle • IBM Db2 • Microsoft SQL Server	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.	
• 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	Currently, data can only be exported from DIS to CSS, Apache Kafka, or DMS Kafka.	For details, see **From DIS**.
• Apache Kafka • DMS Kafka	Currently, data can only be exported from Kafka to CSS, DIS, or DMS Kafka.	For details, see **From Apache Kafka/DMS Kafka**.

Migration Source	Description	Parameter Settings
• Cloud Search Service • Elasticsearch	Data can be exported from CSS or Elasticsearch.	For details, see **From Elasticsearch or CSS**.

Step 6 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 • FusionInsight HDFS • Apache HDFS	You can select a compression format when importing data to HDFS.	For details, see **To HDFS**.
• MRS HBase • FusionInsight HBase • Apache 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**.
• FTP • SFTP • Network Attached Storage • SFS Turbo	When FTP/SFTP/NAS servers function as the migration destination, CDM usually migrates cloud data analysis results back to local file systems.	For details, see **To FTP/ SFTP/NAS/SFS**.
• Data Warehouse Service • RDS for MySQL • RDS for SQL Server • RDS for PostgreSQL	Data can be imported to cloud database services.	For details about how to use the JDBC API to import data, see **To a Relational Database**. • When importing data to DWS, specify the **COPY** or **GDS** import mode to improve the import performance. You can specify the **Import Mode**
FusionInsight LibrA	Data can be imported to FusionInsight LibrA but cannot be imported to Derecho (GaussDB).	

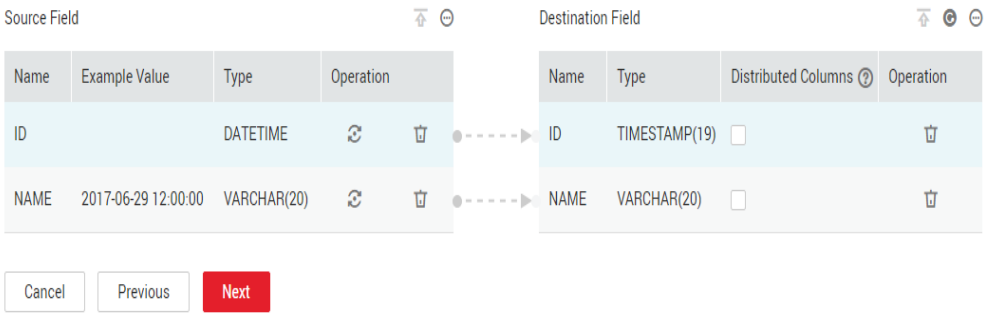
Migration Destination	Description	Parameter Settings
MySQL	The data source can be the on-premises MySQL, MySQL built on ECSs, or MySQL on the third-party cloud.	parameter when **creating a DWS link**. • When importing data to RDS for MySQL, enable the LOAD DATA function of MySQL to accelerate data import and improve the import performance. You can configure **Use Local API** to enable the function when you **create a MySQL link**.
Document Database Service	Data can be imported to the DDS but cannot be imported to the local MongoDB.	For details, see **To DDS**.
Distributed Cache Service	Data can be imported to DCS in the **String** or **Hashmap** value type. Data cannot be imported to the local Redis.	For details, see **To DCS**.
• Cloud Search Service • Elasticsearch	Data can be imported to Elasticsearch or CSS.	For details, see **To Elasticsearch or CSS**.
Data Lake Insight	Data can be imported to DLI.	For details, see **To DLI**.

Step 7 After the parameters are configured, click **Next**. The **Map Field** tab page is displayed.

If files are migrated between FTP, SFTP, NAS, HDFS, and OBS 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

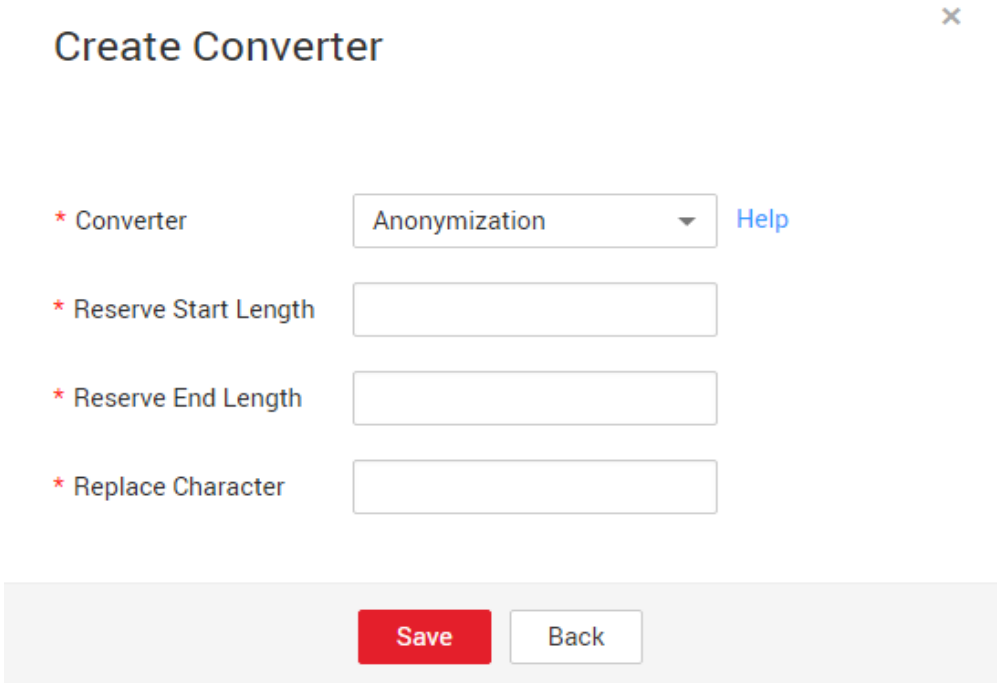


NOTE

- 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, 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.
- 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 8 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.

Step 9 Click **Next**, set job parameters, and click **Show Advanced Attributes** to display and configure optional parameters.

Figure 5-5 Task parameters

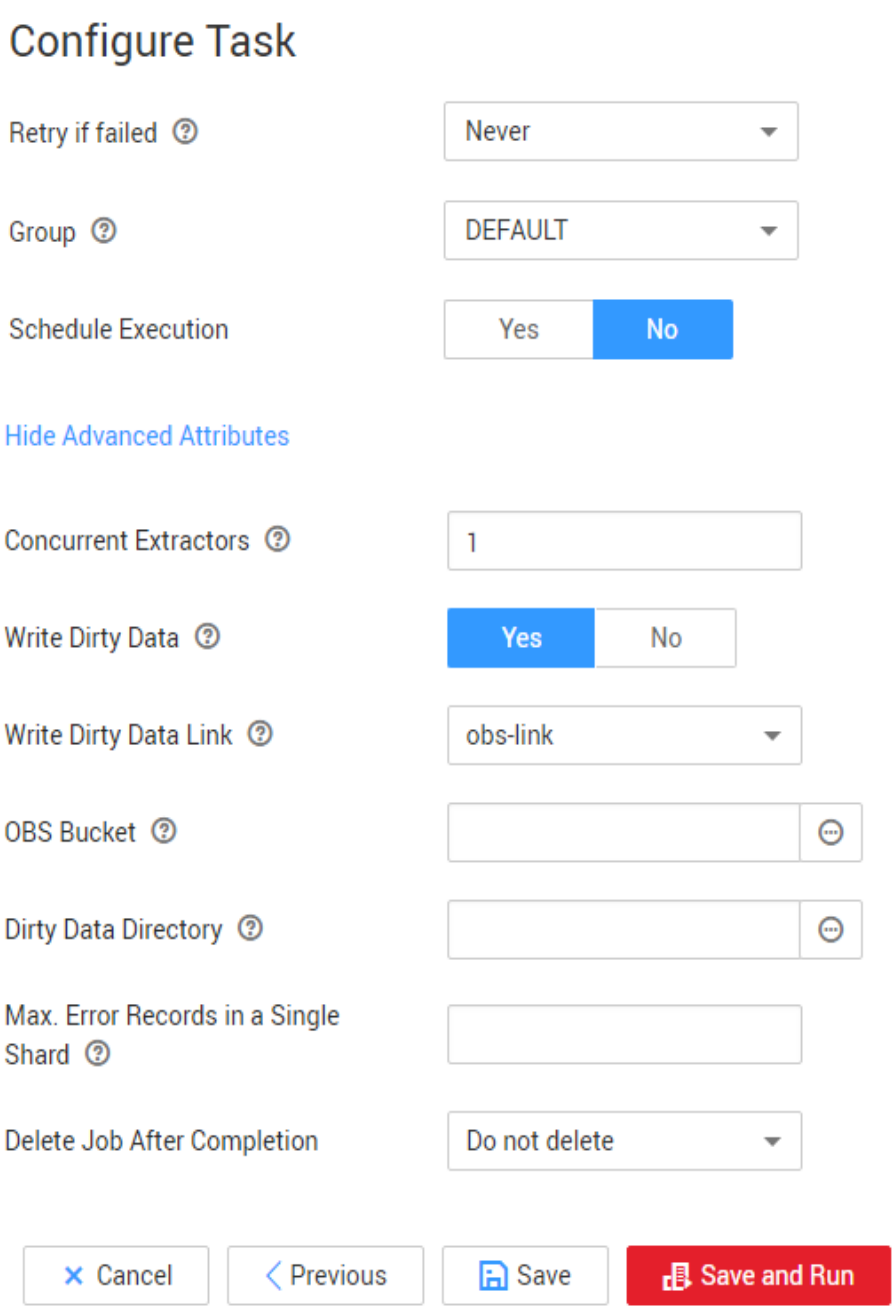


Table 5-3 describes related parameters.

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.	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**.	No
Concurrent Extractors	Number of extractors to be concurrently executed. Generally, retain the default value.	1
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
Write Dirty Data	Whether to record dirty data. By default, this parameter is set to **No**.	Yes
Write Dirty Data Link	This parameter is displayed only when **Write Dirty Data** is set to **Yes**. Only links to OBS support dirty data writes.	obs_link

Parameter	Description	Example Value
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
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

Parameter	Description	Example Value
Delete Job After Completion	After a job is executed, you have three choices: • **Do not delete**: The job is not deleted after it is executed. • **Delete after success**: The job is deleted only when the job is successfully executed. It is used for massive one-time jobs. • **Delete**: The job is deleted regardless of whether it is successfully executed or fails to be executed.	Do not delete

Step 10 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**, or **Succeeded**.

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 Source Job Parameters

5.2.1 From OBS/OSS/KODO/COS/S3

If the source link of a job is the **Link to OBS**, **Link to OSS on Alibaba Cloud**, **Link to KODO/COS**, or **Link to Amazon S3**, configure the source job parameters based on **Table 5-4**.

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

Table 5-4 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Bucket Name	Name of the bucket from which data will be migrated	BUCKET_2

Category	Parameter	Description	Example Value
	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**. 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**.	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
	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
	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 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 line of the CSV file as the heading line and does not write the line 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

Category	Parameter	Description	Example Value
	Compression Format	This parameter is displayed only when **File Format** is set to **CSV** or **JSON**. 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.	NONE
	Source File Processing Method	Operation performed on source files after the job completes. • **Rename**: After the job completes, the source files are renamed by appending usernames and timestamps as suffixes to the file names. • **Delete**: After the job completes, the source files are deleted.	Rename
	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 **|**.	|

Category	Parameter	Description	Example Value
	Wildcard	If you select **Yes**, enter wildcard characters to filter files. All paths or files that meet the search criteria are transferred. For details, see **File/Path Filter**.	Yes
	Filter Type	Only files that meet the filtering conditions are transferred. The options are **None**, **Wildcard**, and **Regex**.	Wildcard
	Directory Filter	If you set **Filter Type** to **Wildcard**, enter a wildcard character to filter paths. The paths that meet the filtering condition are migrated. You can configure multiple paths separated by commas (,).	*input
	File Filter	If you set **Filter Type** to **Wildcard**, 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 (,).	*.csv,*.txt
	Time Filter	If you select **Yes**, files are transferred based on their modification time.	Yes
	Minimum Timestamp	If you set **Filter Type** to **Time Filter**, and specify a point in time for this parameter, only the files modified 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.	2019-06-01 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.	2019-07-01 00:00:00
	Encryption	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
	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
	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 decryption and encryption 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 initialization vectors are inconsistent, the system does not report an exception, but the decrypted data is incorrect.	5C91687BA886EDCD12ACBC3FF19A3C3F
	MD5 File Extension	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.

5.2.2 From HDFS

When the source link of a job is the **Link to HDFS**, that is, when data is exported from MRS HDFS, FusionInsight HDFS, or Apache HDFS, configure the source job parameters based on **Table 5-5**.

Table 5-5 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Source Directory/ File	Directory or file path from which data will be extracted. 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**.	/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
	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 List File	Name of the OBS bucket that stores the text file	obs-cdm-hwstaff

Category	Parameter	Description	Example Value
	OBS Directory	Custom OBS directories that store the text file. Use slashes (/) to separate different directories.	test1
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 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 line of the CSV file as the heading line and does not write the line to the destination table.	No

Category	Parameter	Description	Example Value
	File Split Method	Whether to split files by file or size. If HDFS files are split, each shard is regarded as an individual file. • **File**: Separate files by file quantity. If there are 10 files and **Concurrent Extractors** is set to **5**, each shard consists of two files. • **Size**: Separate files by file size. Files will not be split for balance. Suppose there are 10 files, among which nine are 10 MB and one is 200 MB in size. If **Concurrent Extractors** is set to **2**, two shards will be created, one for processing the nine 10 MB files, the other one for processing the 200 MB file.	File
	Source File Processing Method	Operation performed on source files after the job completes. • **Rename**: After the job completes, the source files are renamed by appending usernames and timestamps as suffixes to the file names. • **Delete**: After the job completes, the source files are deleted.	Rename
	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
	Wildcard	If you select **Yes**, enter wildcard characters to filter files. All paths or files that meet the search criteria are transferred. For details, see **File/Path Filter**.	Yes

Category	Parameter	Description	Example Value
	Path Filter	If you set **Filter Type** to **Wildcard**, enter a wildcard character to filter paths. The paths that meet the filtering condition are migrated. You can configure multiple paths separated by commas (,).	*input
	File Filter	If you set **Filter Type** to **Wildcard**, 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 (,).	*.csv
	Time Filter	If you select **Yes**, files are transferred based on their modification time.	Yes
	Minimum Timestamp	If you set **Filter Type** to **Time Filter**, and specify a point in time for this parameter, only the files modified 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.	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.	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 decryption and encryption 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 initialization vectors 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**.

5.2.3 From HBase/CloudTable

When the source link of a job is the **Link to HBase** or **Link to CloudTable**, that is, when data is exported from MRS HBase, FusionInsight HBase, or Apache HBase, configure the source job parameters based on **Table 5-6**.

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 and CloudTable are 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 5-6 Parameter description

Parameter	Description	Example Value
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**.	TBL_2

Parameter	Description	Example Value
Column Families	(Optional) Column families to which the exported data belongs	CF1&CF2
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.	|
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**.	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**.	2019-02-01 20:00:00

5.2.4 From Hive

If the source link of a job is the **Link to Hive**, configure the source job parameters based on **Table 5-7**.

Table 5-7 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

Parameter	Description	Example Value
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**.	TBL_E
Partition Filter Criteria	This parameter is displayed when you click **Show Advanced Attributes**. You can configure multiple values (separated by spaces) or a range. The time macro function is supported.	• Single/ Multi-value filtering: "${dateformat(yyyyMMdd, -1, DAY)} ${dateformat(yyyyMMdd)}" • Filter by range: "${value} >= ${dateformat(yyyyMMdd, -7, DAY)} && ${value} < ${dateformat(yyyyMMdd)}"

NOTE

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

5.2.5 From FTP/SFTP/NAS/SFS

If the source link of a job is the **Link to an FTP or SFTP Server** or **Link to NAS/SFS**, configure the source job parameters based on **Table 5-8**.

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

Table 5-8 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**. 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**.	/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. It is used for file copy. • **JSON**: Source files will be migrated to tables after being converted to JSON format.	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
	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	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

Category	Parameter	Description	Example Value
	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
	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 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 line of the CSV file as the heading line and does not write the line 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	This parameter is displayed only when **File Format** is set to **CSV** or **JSON**. 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.	None

Category	Parameter	Description	Example Value
	Source File Processing Method	Operation performed on source files after the job completes. • **Rename**: After the job completes, the source files are renamed by appending usernames and timestamps as suffixes to the file names. • **Delete**: After the job completes, the source files are deleted.	Rename
	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
	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 files that meet the filtering conditions are transferred. The options are **None**, **Wildcard**, and **Regex**.	None
	Wildcard	If you select **Yes**, enter wildcard characters to filter files. All paths or files that meet the search criteria are transferred. For details, see **File/Path Filter**.	Yes

Category	Parameter	Description	Example Value
	Directory Filter	If you set **Filter Type** to **Wildcard**, enter a wildcard character to filter paths. The paths that meet the filtering condition are migrated. You can configure multiple paths separated by commas (,).	*input,*out
	File Filter	If you set **Filter Type** to **Wildcard**, 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 (,).	*.csv
	Time Filter	If you select **Yes**, files are transferred based on their modification time.	Yes
	Minimum Timestamp	If you set **Filter Type** to **Time Filter**, and specify a point in time for this parameter, only the files modified 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, **${timestamp(dateformat(yyyy-MM-dd HH:mm:ss,-90,DAY))}** indicates that only files generated within the latest 90 days are migrated.	2019-07-01 00:00:00
	Maximum Timestamp	If you set **Filter Type** to **Time Filter**, and specify a point in time for this parameter, only the files modified before the specified time are transferred. The time format must be *yyyy-MM-dd HH:mm:ss*. This parameter can be set to a macro variable of date and time. For example, **${timestamp(dateformat(yyyy-MM-dd HH:mm:ss))}** indicates that only the files whose modification time is earlier than the current time are migrated.	2019-07-30 00:00:00

Category	Parameter	Description	Example Value
	Encryption	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
	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
	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 decryption and encryption 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 and must be the same as the **IV** configured during encryption. If the initialization vectors are inconsistent, the system does not report an exception, but the decrypted data is incorrect.	5C91687BA886EDCD12ACBC3FF19A3C3F
	MD5 File Extension	Check whether the files extracted by CDM are consistent with source files. For details, see **MD5 Verification**.	.md5

5.2.6 From HTTP/HTTPS

When the source link of a job is the HTTP link, configure the source job parameters based on **Table 5-9**. Currently, data can only be exported from the HTTP/HTTPS URLs.

Table 5-9 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
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 Extension	Extension of the files to be decompressed. This parameter is only displayed when **Compression Format** is not **NONE**. The decompression operation is only performed when the filename 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**.	|
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

Parameter	Description	Example Value
Encryption	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 decryption and encryption 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 and must be the same as the **IV** configured during encryption. If the initialization vectors are inconsistent, the system does not report an exception, but the decrypted data is incorrect.	5C91687BA886EDCD12ACBC3FF19A3C3F
MD5 File Extension	Check whether the files extracted by CDM are consistent with source files. For details, see **MD5 Verification**.	.md5

5.2.7 From a Relational Database

When the source link of a job is one of the relational databases listed in **Link to Relational Databases** (also listed here), configure the source job parameters based on **Table 5-10**.

- Data Warehouse Service
- RDS for MySQL
- RDS for SQL Server
- RDS for PostgreSQL
- Dameng database
- FusionInsight LibrA

- Derecho (GaussDB)
- MySQL
- PostgreSQL
- Oracle
- IBM Db2
- Microsoft SQL Server

Table 5-10 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
	SQL Statement	When **Use SQL Statement** is set to **Yes**, enter an SQL statement here. CDM exports data based on the SQL statement.	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, confirm that the table exists or that the login account has the permissions required 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** 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
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.	id

Category	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. This parameter can be configured as a macro variable of date and time to extract data generated at a specific date. For details, see **WHERE Clause**.	DS='${dateformat(yyyy-MM-dd,-1,DAY)}'
	Null in Partition Column	Whether the partition column can contain null values	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
	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.	Yes
	Job Split Field	Used to split a job into multiple subjobs for concurrent execution.	-
	Minimum value of a split field	Specifies the minimum value of **Job Split Field** during data extraction.	-

Category	Parameter	Description	Example Value
	Maximum Split Field Value	Specifies the maximum value of **Job Split Field** during data extraction.	-
	Number of subjobs	Specifies the number of subjobs split from a job based on the data range specified by the minimum and maximum values of **Job Split Field**.	-

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.
- In a migration from MySQL to DWS, the constraints on the incremental data migration function in MySQL Binlog mode are as follows:
  - A single cluster supports only one incremental migration job in MySQL Binlog mode in the current version.
  - In the current version, you are not allowed to delete or update 10,000 data records at a time.
  - Entire DB migration is not supported.
  - Data Definition Language (DDL) operations are not supported.
  - Event migration is not supported.
  - If you set **Migrate Incremental Data** to **Yes**, **binlog_format** in the source MySQL database must be set to **ROW**.
  - If you set **Migrate Incremental Data** to **Yes** and binlog file ID disorder occurs on the source MySQL instance due to cross-machine migration or rebuilding during incremental data migration, incremental data may be lost.
  - If a primary key exists in the destination table and incremental data is generated during the restart of the CDM cluster or full migration, duplicate data may exist in the primary key. As a result, the migration fails.
  - If the destination DWS database is restarted, the migration will fail. In this case, restart the CDM cluster and the migration job.
- The recommended MySQL configuration is as follows:

```
# Enable the bin-log function.
log-bin=mysql-bin
# ROW mode
binlog-format=ROW
# gtid mode. The recommended version is 5.6.10 or later.
gtid-mode=ON
enforce_gtid_consistency = ON
```


Table 5-11 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. **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	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, confirm that the table exists or that the login account has the permissions required 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** 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
Advanced Attributes	Where Clause	Specifies the data extraction range. If this parameter is not set, the entire table is extracted. This parameter can be configured as a macro variable of date and time to extract data generated at a specific date. For details, see **WHERE Clause**.	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)}

5.2.8 From MongoDB/DDS

When you migrate data from MongoDB or DDS to a relational database, 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.

When the source link of a job is the **Link to MongoDB**, that is, when data is exported from an on-premises MongoDB or DDS, configure the source job parameters based on **Table 5-12**.

Table 5-12 Parameter description

Parameter	Description	Example Value
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

Parameter	Description	Example Value
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'}

5.2.9 From Redis

Because DCS restricts the commands for obtaining keys, it cannot serve as the migration source but can be the migration destination. 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.

When data is exported from an on-premises Redis, configure source job parameters as described in **Table 5-13**.

Table 5-13 Parameter description

Parameter	Description	Example Value
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
Key Delimiter	Character used to separate table names and column names of a relational database	_

Parameter	Description	Example Value
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

5.2.10 From DIS

Currently, data can only be exported from DIS to CSS, Apache Kafka, or DMS Kafka.

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 the **Link to DIS**, configure the source job parameters based on **Table 5-14**.

Table 5-14 Parameter description

Parameter	Description	Example Value
DIS Stream	DIS stream name	dis
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
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

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.	Binary
Field Delimiter	The default value is space. To set the **Tab** key as the delimiter, set this parameter to **\t**.	,
Max. Poll Records	(Optional) Maximum number of records per poll	100
Application Name	Unique identifier of the consumer application to be used. If no application exists, CDM creates one automatically.	cdm

5.2.11 From Apache Kafka/DMS Kafka

Currently, data can only be exported from Kafka to CSS, DIS, or DMS Kafka.

If the source link of a job is the **Link to Kafka** or **Link to DMS Kafka**, configure the source job parameters based on **Table 5-15**.

Table 5-15 Parameter description

Parameter	Description	Example Value
Topics	One or more topics can be entered.	est1,est2
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.	Latest
Permanent Running	Whether a job runs permanently.	Yes
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

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.	Binary
Field Delimiter	The default value is space. To set the **Tab** key as the delimiter, set this parameter to **\t**.	,
Max. Poll Records	(Optional) Maximum number of records per poll	100
Max. Poll Interval	(Optional) Maximum interval between polls (seconds)	100

5.2.12 From Elasticsearch or CSS

If the source link of a job is the **Link to Elasticsearch/CSS**, configure the source job parameters based on **Table 5-16**.

Table 5-16 Parameter description

Parameter	Description	Example Value
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.	type
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
Filter Conditions	(Optional) Whether to use a query string to filter the source data. CDM only migrates the data that meets the filtering conditions.	last_name:Smith
Extract Meta-field	Whether to extract index meta-fields. For example, _index, _type, _id, and _score.	Yes

5.2.13 From OpenTSDB

If the source link of a job is the **Link to CloudTable OpenTSDB**, configure the source job parameters based on **Table 5-17**.

Table 5-17 Parameter description

Parameter	Description	Example Value
Start Time	Start time of the query. The value is a character string or timestamp in the format of *yyyyMMddHHmmdd*.	20180920145505
End Time	(Optional) End time of the query. The value is a string or timestamp in the format of *yyyyMMddHHmmdd*.	1598870800
Metric	Metric of the data to be migrated. You can specify a metric or select an existing metric in OpenTSDB.	city.temp
Aggregate Function	Aggregate function	sum
Tag	(Optional) If you specify a tag, only the tagged data will be migrated.	tagk1:tagv1,tagk2:tagv2

5.3 Destination Job Parameters

5.3.1 To OBS

If the destination link of a job is the **Link to OBS**, configure the destination job parameters based on **Table 5-18**.

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

Table 5-18 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Bucket Name	Name of the OBS bucket that data will be written to	bucket_2

Category	Parameter	Description	Example Value
	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**.	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, NAS, HDFS, and OBS, the value of **File Format** must the same as the source file format.	CSV
	Duplicate File Processing Method	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 **Skipping Duplicate Files**.	Skip

Category	Parameter	Description	Example Value
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. • **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**.	KMS
	Key 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
	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 initialization vectors are inconsistent, the system does not report an exception, but the decrypted data is incorrect.	5C91687BA886EDCD12ACBC3FF19A3C3F
	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**.	,

Category	Parameter	Description	Example Value
	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
	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

Category	Parameter	Description	Example Value
	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 **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
	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.	${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 **sqltabname.csv**. • Macro variable of version number: If this parameter is set to **${version}**, the name of the generated file is **v1.csv**. • Any combination of the character string and macro variable (macro variable of time, table name, or version number). For example, if this parameter is set to **cdm#${timestamp()}_${version}**, the name of the generated file is **cdm#1554108737_v1.csv**.	cdm

5.3.2 To HDFS

If the destination link of a job is one of them listed in **Link to HDFS**, configure the destination job parameters based on **Table 5-19**.

Table 5-19 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**.	/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, NAS, HDFS, and OBS, the value of **File Format** must the same as the source file format.	CSV
Duplicate File Processing Method	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
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

Parameter	Description	Example Value
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	This parameter is displayed only when **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
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
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 initialization vectors 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**.

5.3.3 To HBase/CloudTable

If the destination link of a job is one of them listed in **Link to HBase** or **Link to CloudTable**, configure the destination job parameters based on **Table 5-20**.

- MRS HBase
- FusionInsight HBase

- Apache HBase
- CloudTable Service

Table 5-20 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**.	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
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

Parameter	Description	Example Value
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
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

5.3.4 To Hive

If the destination link of a job is the **Link to Hive**, configure the destination job parameters based on **Table 5-21**.

Table 5-21 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

Parameter	Description	Example Value
Auto Table Creation	This parameter is displayed only when both the migration source and destination are relational databases. 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	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**.	TBL_X
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

NOTE

1. When Hive serves as the migration destination, the storage format selected during table creation will be automatically used, such as ORC and Parquet.
2. When Hive serves as the migration destination, if the storage format is TEXTFILE, delimiters must be explicitly specified in the statement for creating Hive tables. The following gives 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;
```

5.3.5 To FTP/SFTP/NAS/SFS

If the destination link of a job is the **Link to an FTP or SFTP Server** or **Link to NAS/SFS**, configure the destination job parameters based on **Table 5-22**.

Table 5-22 Parameter description

Category	Parameter	Description	Example Value
Basic parameters	Write Directory	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**.	/opt/ftp/

Category	Parameter	Description	Example Value
	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, NAS, HDFS, and OBS, the value of **File Format** must the same as the source file format.	CSV
	Duplicate File Processing Method	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	Skip
Advanced attributes	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
	Encoding Type	Encoding type, for example, **UTF-8** or **GBK**. This parameter is not used when **File Format** is set to **Binary**.	GBK

Category	Parameter	Description	Example Value
	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
	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
	Generate MD5 Hash Value	This parameter is displayed only when the migration source is a file system (OBS/FTP/SFTP/NAS/HDFS), the migration destination is FTP/SFP/NAS, and **File Format** is **Binary**. 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.	No
	Directory of MD5 Hash Value	Directory for storing MD5 values	/md5
	Use First Row as Header	This parameter is displayed only when **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
	Encryption	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 initialization vectors are inconsistent, the system does not report an exception, but the decrypted data is incorrect.	5C91687BA886EDCD12ACBC3FF19A3C3F

5.3.6 To a Relational Database

If the destination link of a job is one of them listed in **Link to Relational Databases**, configure the destination job parameters based on **Table 5-23**.

- DWS
- RDS for MySQL
- RDS for SQL Server
- RDS for PostgreSQL

- FusionInsight LibrA
- MySQL

Table 5-23 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
Auto Table Creation	This parameter is displayed only when both the migration source and destination are relational databases. 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.	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**.	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 condition columns and row store indexes are uncertain).	Row-based
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	Used to specify the data to be deleted from the destination table before data import	age > 18 and age <= 60
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 set **Clear Data Before Import** to **Yes**, CDM does not roll back the deleted data even in transaction mode.	No

Parameter	Description	Example Value
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. 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

Field Mapping in Automatic Table Creation on DWS

Figure 5-6 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 5-6 Field mapping in automatic table creation on DWS

Source Database							Destination Database
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	TIME	BOOLEAN	BOOLEAN	BOOLEAN	BOOLEAN	BOOLEAN

5.3.7 To DDS

If the destination link of a job is the **Link to MongoDB**, configure the destination job parameters based on **Table 5-24**.

Table 5-24 Parameter description

Parameter	Description	Example Value
Database Name	Database to which data is to be imported	mongodb
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

5.3.8 To DCS

If the destination link of a job is the **Link to Redis/DCS**, that is, when data is imported to DCS, configure the destination job parameters based on **Table 5-25**.

Table 5-25 Parameter description

Parameter	Description	Example Value
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
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	;

5.3.9 To Elasticsearch or CSS

If the destination link of a job is the **Link to Elasticsearch/CSS**, that is, when data is imported to Elasticsearch or CSS, configure the destination job parameters based on **Table 5-26**.

Table 5-26 Parameter description

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.	type
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

Parameter	Description	Example Value
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

5.3.10 To DLI

If the destination link of a job is the **Link to DLI**, configure the destination job parameters based on **Table 5-27**.

Table 5-27 Parameter description

Parameter	Description	Example Value
Resource Queue	Resource queue to which the destination table belongs	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

Parameter	Description	Example Value
Clear Data Before Import	Whether to clear data in the destination table before data import	No

5.3.11 To DIS

If the destination link of a job is the **Link to DIS**, configure the destination job parameters based on **Table 5-28**.

Table 5-28 Parameter description

Parameter	Description	Example Value
DIS Stream	DIS stream name	cdm
Field Delimiter	The default value is space. To set the **Tab** key as the delimiter, set this parameter to **\t**.	,
Reference Sign	Delimiter between the names of the referenced tables or columns. This parameter is left blank by default.	'

5.3.12 To OpenTSDB

If the destination link of a job is the **Link to CloudTable OpenTSDB**, configure the destination job parameters based on **Table 5-29**.

Table 5-29 Parameter description

Parameter	Description	Example Value
Metric	(Optional) You can specify a metric or select an existing metric in OpenTSDB.	city.temp
Time	(Optional) Data point. The value is a string or timestamp in the format of *yyyyMMddHHmmdd*.	1598870800
Tag	(Optional) Data tag	tagk:tagv, tagk2:tagv2

5.4 Entire DB Migration

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 or each collection of MongoDB can be executed concurrently as a subtask.

Table 5-30 lists the data sources supporting entire DB migration using .

Table 5-30 Supported data sources in entire DB migration

Source Data Type	Destination Data Type					
	RDS for MySQL	MRS (Hive)	DWS	CSS	OBS	CloudTable
MySQL	√	√	√	×	√	×
PostgreSQL	√	√	√	×	√	×
Microsoft SQL Server	√	√	√	×	×	×
Oracle	√	√	√	×	√	×
Elasticsearch	×	×	×	√	×	×
MongoDB	×	×	×	×	×	×
HBase	×	×	×	×	×	√
IBM Db2	√	√	√	×	√	×
Derecho (GaussDB)	√	√	√	×	√	×
SAP HANA	√	√	√	×	√	×
DWS	√	√	√	×	×	×
Hive	√	×	√	×	×	×

The source databases can be deployed in on-premises data centers or built on ECSs, or third-party database services.

Field Mapping in Automatic Table Creation

CDM automatically creates tables at the destination during database migration. **Figure 5-7** describes the field mapping between the 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 5-7 Field mapping in automatic table creation on DWS

Source Database							Destination Database
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	TIME	BOOLEAN	BOOLEAN	BOOLEAN	BOOLEAN	BOOLEAN

Prerequisites

- You have created links according to **Creating Links**.
- The CDM cluster can communicate with the data source.

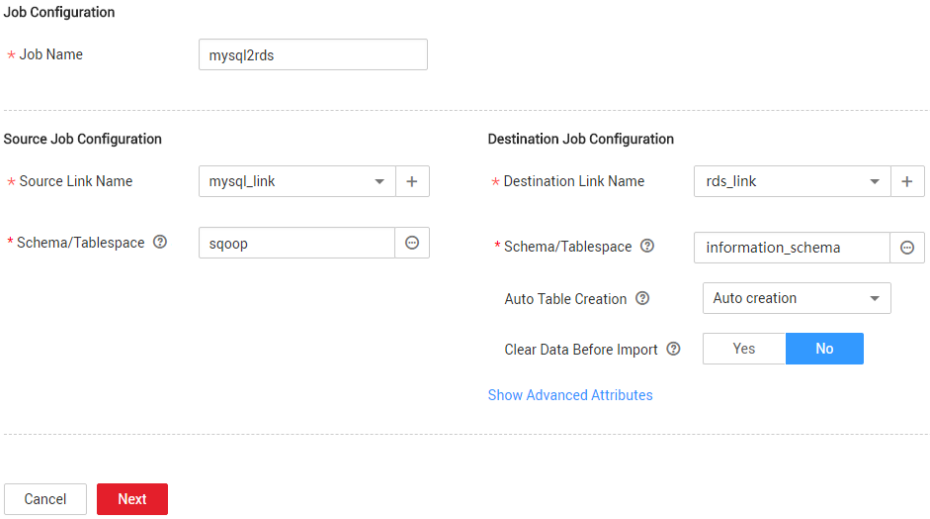
Procedure

Step 1 Log in to the **CDM management console**.

Step 2 In the left navigation pane, click **Cluster Management**. Locate the target cluster and click **Job Management**.

Step 3 Choose **Entire DB Migration** > **Create Job**. The page for configuring the job is displayed.

Figure 5-8 Creating an entire DB migration job



Step 4 Configure the related parameters of the source database according to **Table 5-31**.

Table 5-31 Parameter description

Source Database	Parameter	Description	Example Value
• Oracle • MySQL • PostgreSQL • Microsoft SQL Server	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

Source Database	Parameter	Description	Example Value
	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. This parameter can be configured as a macro variable of date and time to extract data generated at a specific date. For details, see **WHERE Clause**.	age > 18 and age <= 60
Elasticsearch	Index	Index of the data to be extracted. The value can be a wildcard character. Multiple indexes that meet the wildcard condition can be migrated at a time. 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. If multiple indexes are migrated at the same time, **Index** cannot be configured at the migration destination.	cdm*
MongoDB	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

Step 5 Configure the related parameters, from **Table 5-32**, for the destination cloud service.

Table 5-32 Parameter description

Cloud Service	Parameter	Description
• MRS Hive • RDS for MySQL	Schema/ Tablespace	Database name
	Auto Table Creation	The options are as follows: • **Non-auto creation**: CDM will not automatically create a table. • **Auto creation**: If no corresponding table exists in the destination database, CDM will automatically create one. • **Deletion before creation**: If a table with the same name exists in the destination database, CDM will delete the table first and create another one with the same name.
	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 the data to be deleted from the destination table.
	WHERE Clause	Used to specify the data to be deleted from the destination table before data import, for example, **age > 18 and age <= 60**

Cloud Service	Parameter	Description
CSS	Index	Index to which data is written. 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	Whether to clear data of the target type before data is written
DWS	-	For details about the destination job parameters required for entire DB migration to DWS, see **To a Relational Database**.
OBS	-	For details about the destination job parameters required for entire DB migration to OBS, see **To OBS**.

Step 6 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 7 Click **Next** and set job parameters.

Table 5-33 describes related parameters.

Table 5-33 Task configuration parameters

Parameter	Description	Example Value
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 8 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

5.5 Scenario-based Migration

Scenario-based migration migrates snapshots and then restores table data to speed up migration.

Prerequisites

- The CDM cluster can communicate with the data source.
- 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.

Link to Hadoop

CDM supports the following Hadoop data sources:

MRS

When connecting CDM to Hadoop of MRS, configure the parameters as described in **Table 5-34**.

Table 5-34 MRS Hadoop 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_scen_link
Manager IP	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**: for non-security mode ● **KERBEROS**: for security mode	SIMPLE
HBase Version	Set it to the HBase version on the server.	HBASE_2_X
HIVE Version	Set it 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.	cdm
Password	Password used for logging in to MRS Manager	-
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. ● **Agent**: The link instance runs on an agent. If **STANDALONE** is selected, CDM can migrate data between HDFSs of multiple MRS clusters.	STANDALONE

FusionInsight Hadoop

When connecting CDM to Hadoop of FusionInsight HD, configure the parameters as described in **Table 5-35**.

Table 5-35 FusionInsight Hadoop 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 FusionInsight HD • **SIMPLE**: for non-security mode • **KERBEROS**: for security mode	KERBEROS
HBase Version	Set it to the HBase version on the server.	HBASE_2_X
HIVE Version	Set it to the Hive version on the server.	HIVE_3_X

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

Apache Hadoop

When connecting CDM to Apache Hadoop, configure parameters as described in **Table 5-36**.

Table 5-36 Apache Hadoop 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	hdfs://nn1.example.com/
ZooKeeper Address	ZooKeeper address, which needs to be configured for HBase scenario-based migration	hbase-node-1:2181
Hive Metastore	Hive metadata address. For details, see the **hive.metastore.uris** configuration item.	thrift://host-192-168-1-212:9083

Parameter	Description	Example Value
Authentication Method	Authentication method used for accessing Hadoop • **SIMPLE**: Select this if Hadoop is in non-security mode. • **KERBEROS**: Select this if Hadoop is in security mode. Obtain the **Principal** account and **Keytab File** file of the client for authentication.	KERBEROS
Principal	When **Authentication Method** is set to **KERBEROS**, the **Principal** account is used for authentication. You can contact the Hadoop administrator to obtain the account.	USER@YOUR-REALM.COM
Keytab File	When **Authentication Method** is set to **KERBEROS**, this file is used for authentication. You can contact the Hadoop administrator to obtain the file.	/opt/user.keytab
IP and Host Name Mapping	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
HBase Version	Set it to the HBase version on the server.	HBASE_2_X
HIVE Version	Set it to the Hive version on the server.	HIVE_3_X

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

Procedure

Step 1 Log in to the **CDM management console**.

Step 2 In the left navigation pane, click **Cluster Management**. Locate the target cluster and click **Job Management**.

Step 3 Choose **Job Management** > **Link Management** > **Create Link** and set the connector type to **Hadoop release version**.

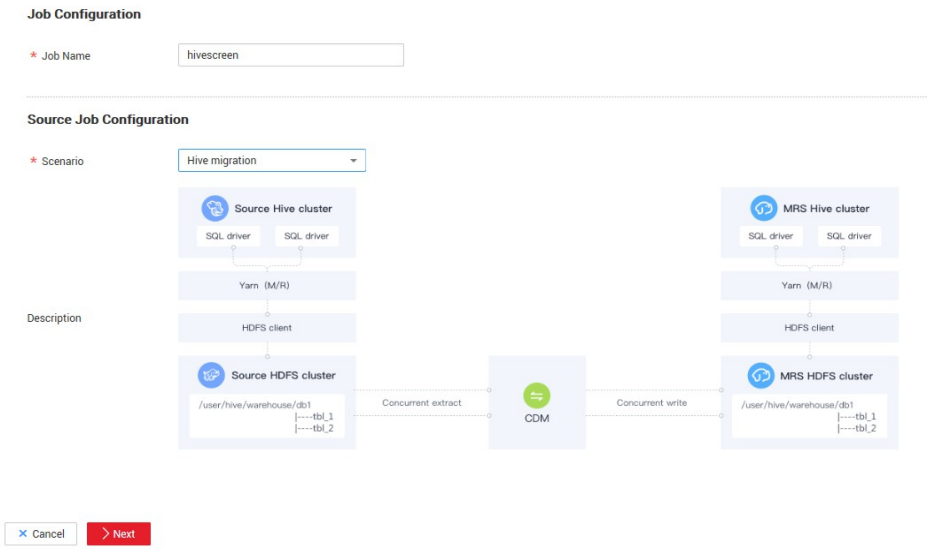
Step 4 Click **Next**. Set link parameters by referring to **Link to Hadoop**.

Step 5 Click **Test** to check whether the link is available. Alternatively, click **Save**. The system will automatically check whether the link is available.

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

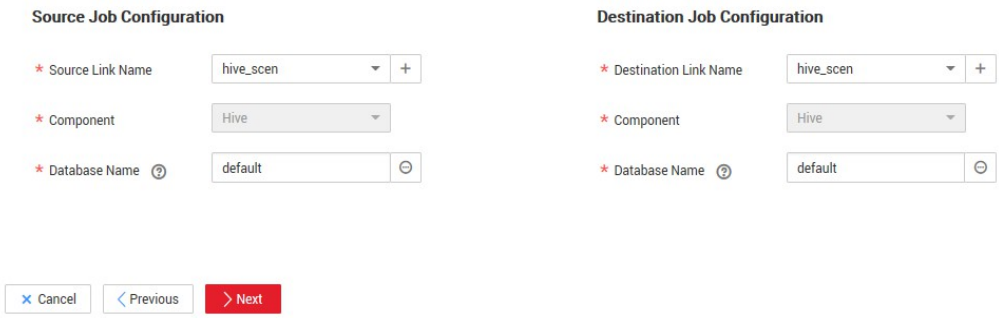
Step 6 Choose **Scenario Migration** > **Create Job**. The page for configuring the job is displayed. Select a migration scenario (Hadoop migration, Hive migration, or HBase migration) and configure the job name.

Figure 5-9 Configuring a scenario-based migration Job



Step 7 Configure the source and destination job parameters, and select the link name and name of the database to be migrated.

Figure 5-10 Configuring job parameters



Step 8 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 9 Click **Next** and set job parameters.

Table 5-37 describes related parameters.

Table 5-37 Task configuration parameters

Parameter	Description	Example Value
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

Parameter	Description	Example Value
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
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 10 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

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.

Scheduling Job Execution by Minute

CDM allows jobs to be executed every several minutes. It is recommended that the cycle be at least 10 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 5-11 Scheduling job execution by minute

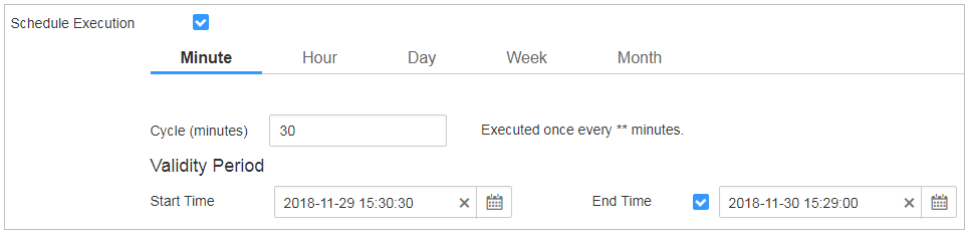


Figure 5-11 shows that the job will be automatically executed at 15:30:30 on November 29, 2018 for the first time at a cycle of 30 minutes, and will be automatically stopped at 15:29:00 on November 30, 2018.

Scheduling Job Execution by Hour

CDM allows jobs to be executed every several hours.

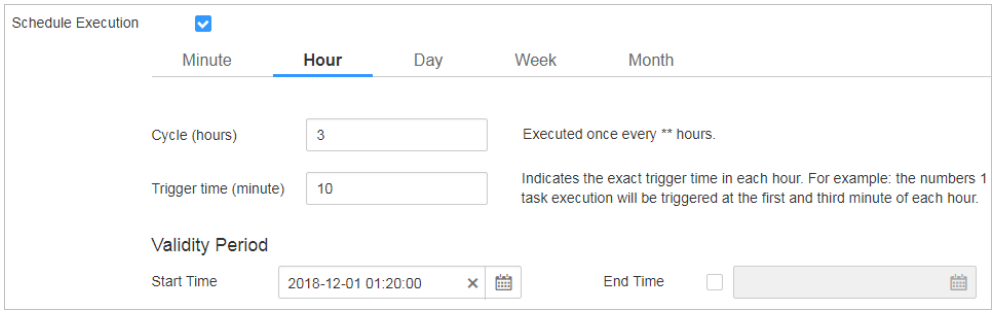
- **Cycle (hours)**: indicates the interval when a job is automatically executed.
- **Trigger Time (minute)**: indicates the exact time in each hour when a scheduled task is triggered. The value ranges from 0 to 59. You can set a maximum of 60 values and use commas (,) to separate these values. However, 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**: **1:20:00**
  - **Cycle (hours)**: **3**
  - **Trigger Time (minute)**: **10**

  Figure 5-12 shows that the first automatic execution time is **2:10:00**, and the second automatic execution time is **5:10:00**.

  Figure 5-12 Trigger time beyond the validity period

- **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 5-13 Scheduling job execution by hour

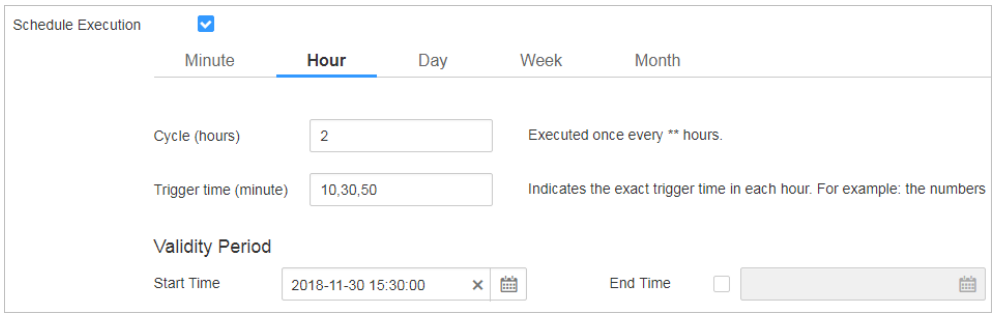


Figure 5-13 shows that the scheduled configuration will take effect at 15:30:00 on November 30, 2018. The job is automatically executed for the first time upon the scheduled configuration takes effect, at 15:50:00 for the second time, and at 17:10:00 for the third time. The job is triggered for three times every 2 hours and the configuration is always valid.

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 5-14 Scheduling job execution by day

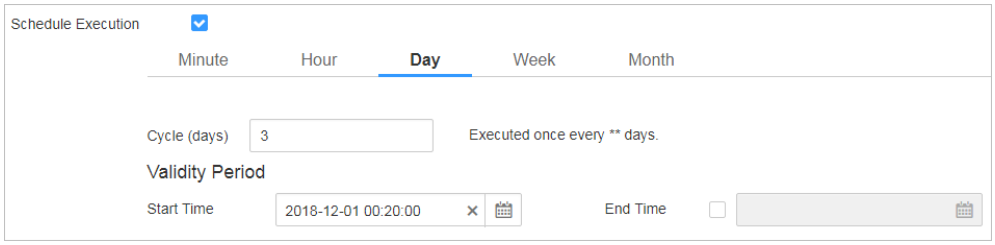


Figure 5-14 shows that the scheduled job will be automatically executed at 00:20:00 on December 1, 2018, and is executed once every three days. The configuration is always valid.

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 5-15 Scheduling job execution by week

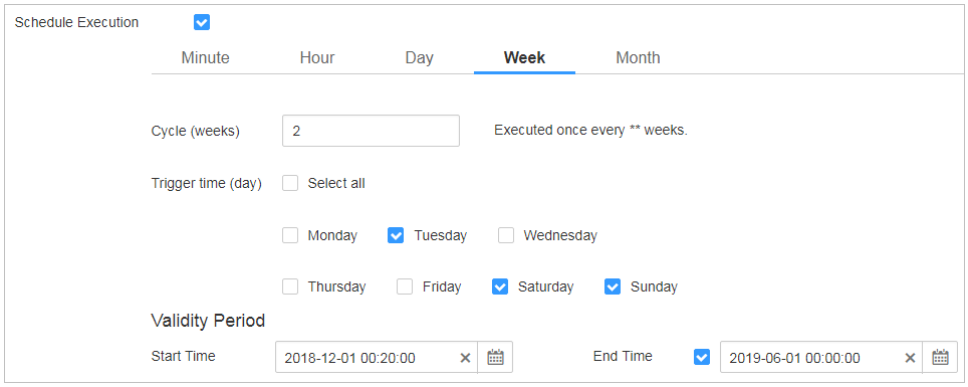


Figure 5-15 shows that the job will be automatically executed at 00:20:00 every Tuesday, Saturday, and Sunday every two weeks starting from 00:20:00 on December 1, 2018, and the job will be automatically stopped at 00:00:00 on June 1, 2019.

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 5-16 Scheduling job execution by month

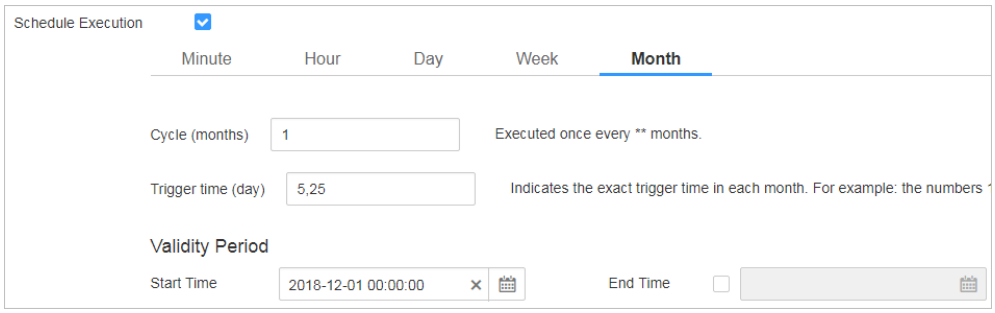


Figure 5-16 shows that the job will be automatically executed at 00:00:00 on every fifth and twenty-fifth day of each month starting from 00:00:00 on December 1, 2018. The configuration is always valid.

5.7 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**, or **Succeeded**.

  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.

- **Viewing historical jobs**

  CDM stores the jobs executed in the last month, including one-time jobs (jobs that are automatically deleted after execution) and jobs that are executed periodically. You can view and re-execute the jobs on the **Historical Jobs** tab page.

For a job that is executed periodically, a historical job is generated on the **Historical Jobs** tab page each time when the job is executed, regardless of whether the job is executed successfully. The names of historical jobs will be the same as the original job but with a random character string appended.

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 **CDM management console**.

Step 2 In the left navigation pane, click **Cluster Management**. Locate the target cluster and click **Job Management**.

Step 3 Click **Historical Jobs** to view all historical jobs executed in the latest month.

CDM stores the jobs executed in the last month, including one-time jobs (jobs that are automatically deleted after execution) and jobs that are executed periodically. You can view and re-execute the jobs on the **Historical Jobs** tab page.

For a job that is executed periodically, a historical job is generated on the **Historical Jobs** tab page each time when the job is executed, regardless of whether the job is executed successfully. The names of historical jobs will be the same as the original job but with a random character string appended.

Step 4 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.

Step 5 After the modification, click **Save** or **Save and Run**.

----End

5.8 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 **CDM management console**.

Step 2 In the left navigation pane, click **Cluster Management**. Locate the target cluster and click **Job Management**.

Step 3 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** to export all jobs in JSON format. These files can be used as backups or imported to another cluster.

  Currently, you cannot select specific jobs to export but can only export all jobs at a time. For security purposes, the link passwords are not exported when CDM export the jobs and are replaced with *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.

----End

6 Job Configuration Management

On the **Configuration Management** tab page, you can perform the following operations:

- **Automatic Backup and Restoration of CDM Jobs**
- **Global Variables of CDM Job Parameters**

Automatic Backup and Restoration of CDM Jobs

- Prerequisites

  You have created the **Link to OBS**.

- Automatic backup

  On the **Job Management** page, click the **Configuration Management** tab and configure the parameters of **Scheduled Backup**.

  Table 6-1 Parameters for **Scheduled Backup**

Parameter	Description	Example Value
Scheduled Backup	Whether to enable automatic backup. This function is used to back up jobs but not links. If this function is enabled, you need to configure **Backup Policy**, **Backup Cycle**, **OBS Link for Writing Backups**, **OBS Bucket**, and **Backup Data Directory**.	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

Parameter	Description	Example Value
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	CDM uses the link to back up jobs to OBS. You must create an OBS link on the **Link Management** tab page in advance.	obslink
OBS Bucket	OBS bucket where backup files are stored	cdm
Backup Data Directory	Directory where backup files are stored	/cdm-bk/
Environment Variable	Global control parameter. For example, if xxx=123 exists, you can use ${xxx} to replace 123 in jobs.	AAA=333
Maximum Concurrent Extractors	It is user-defined. Do not set the same scheduled time for different jobs. Instead, set different times to avoid exceptions.	5

- 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.

Global Variables of CDM 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.

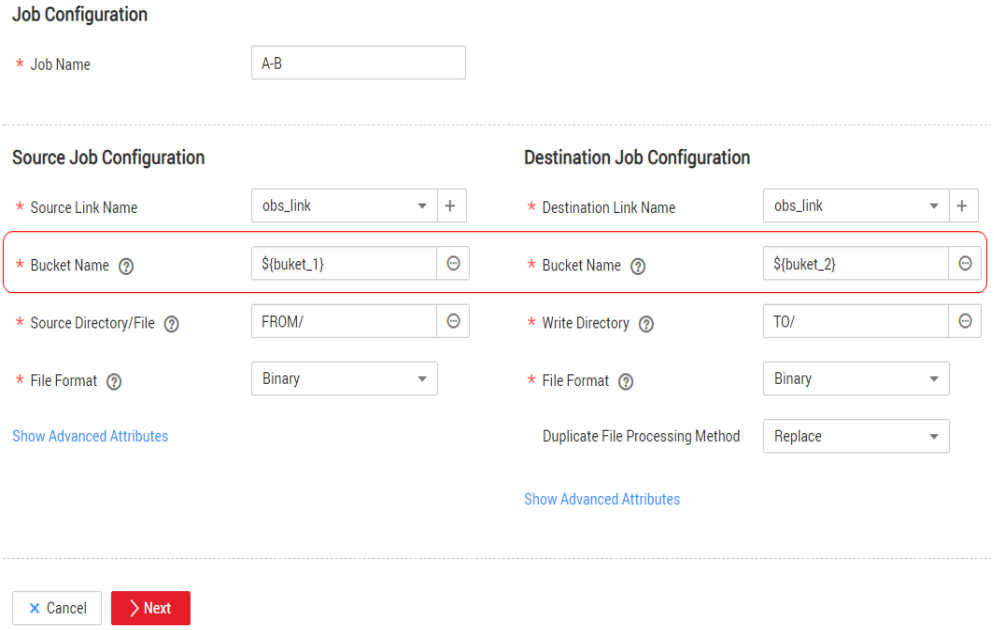
1. On the **Job Management** page, click the **Configuration Management** tab and configure environment variables.

   ```
   buket_1=A
   buket_2=B
   ```

   Variable **buket_1** indicates bucket A, and variable **buket_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 **${buket_1}** and destination bucket name to **${buket_2}**.

Figure 6-1 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:

```
buket_1=C
buket_2=D
```


7 Agent Management

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 7-1 Scenario

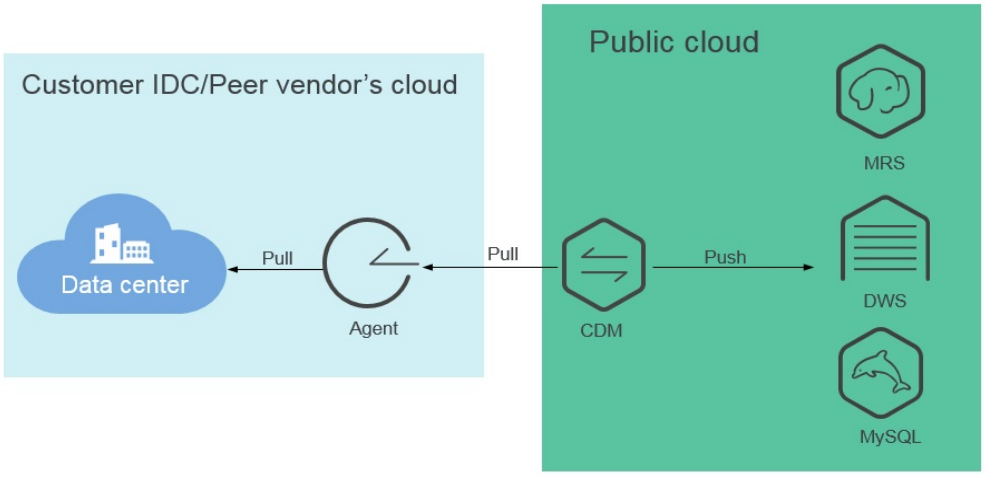
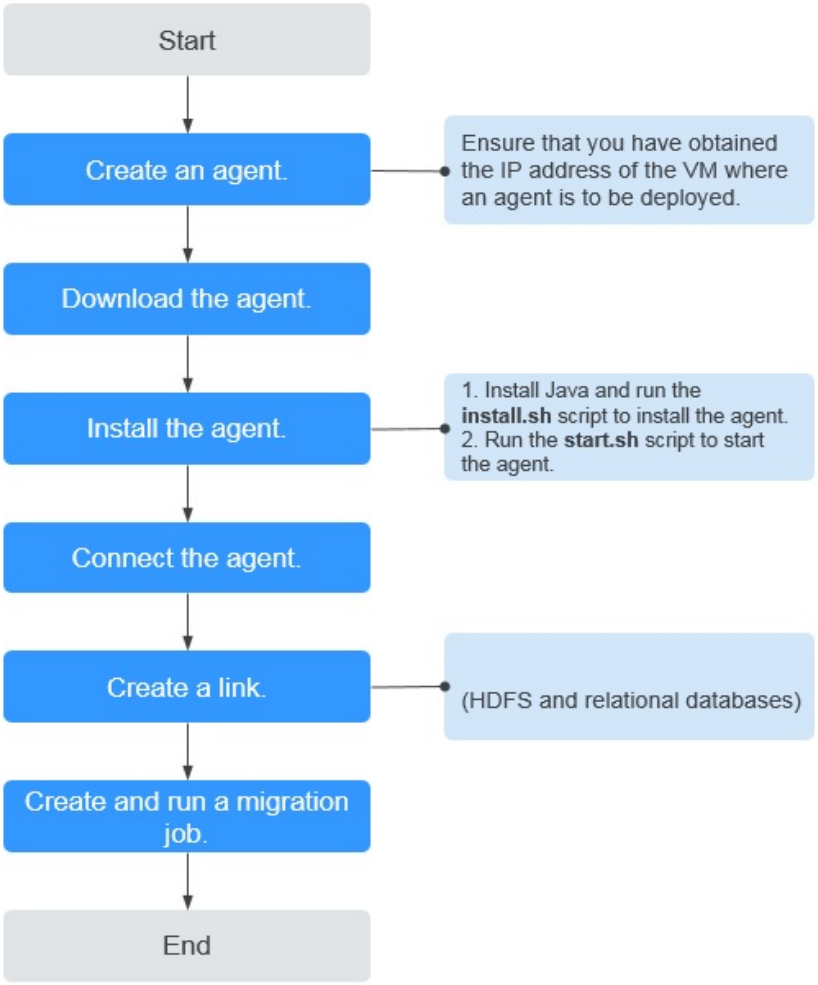


Figure 7-2 shows the process of using an agent.

Figure 7-2 Process



Prerequisites

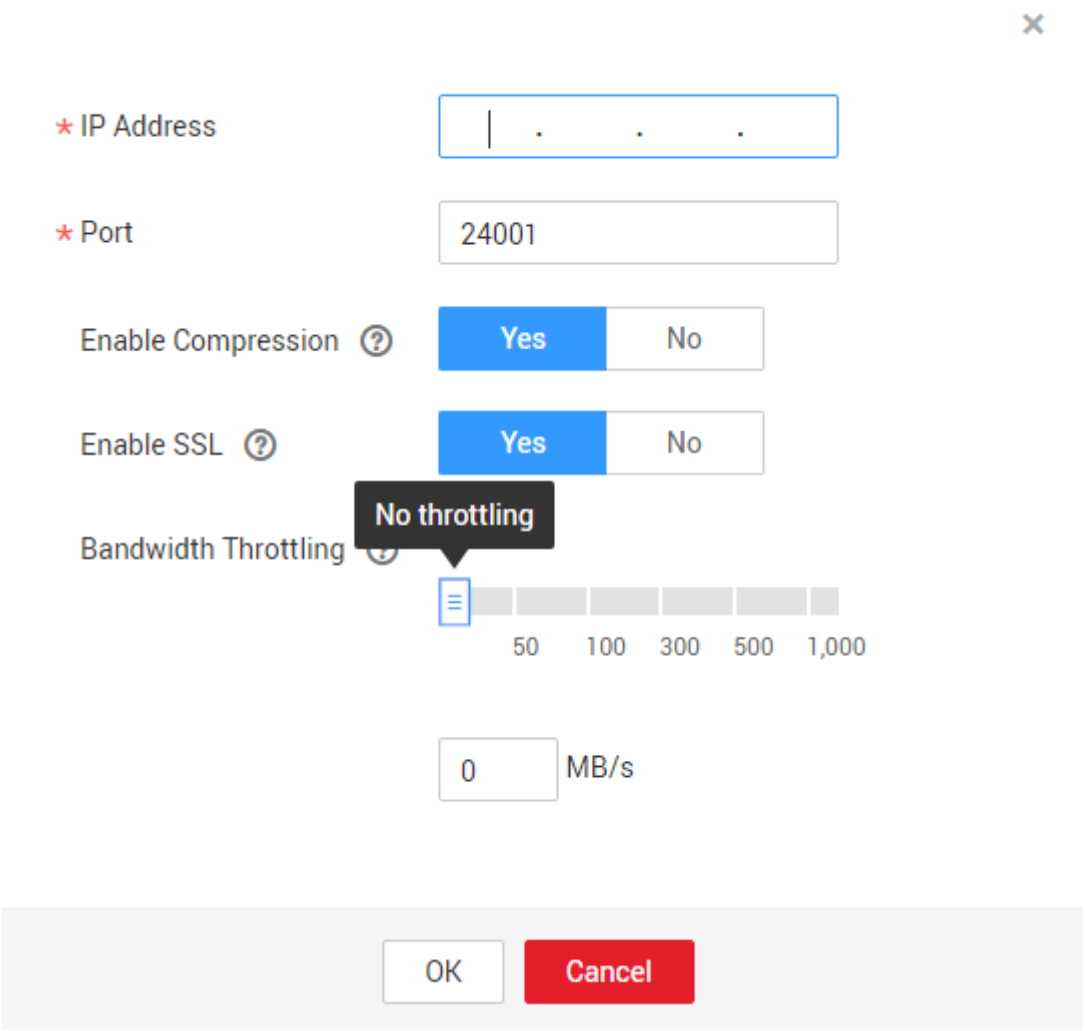
You have created a cluster according to **Creating a CDM Cluster**.

Creating an Agent

Step 1 Log in to the **CDM management console**.

Step 2 In the left navigation pane, click **Cluster Management**. Locate the target cluster, choose **Job Management** > **Agent Management** > **Create Agent**, and configure agent parameters. See **Figure 7-3**.

Figure 7-3 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**: port exposed by the agent. Recommended value range: 1024–65535.
- **Enable Compression**: whether to compress data for transmission
- **Enable SSL**: whether to enable two-way SSL authentication
- **Bandwidth Throttling**: set the maximum downstream rate of the agent. By default, there is no throttling.

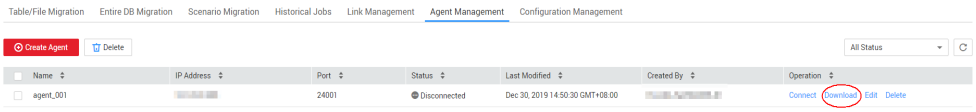
Step 3 Click **OK**. On the **Agent Management** page, view the created agent.

----End

Downloading and Installing an Agent

Step 1 On the **Agent Management** page, locate the created agent. Click **Download** in the **Operation** column. See **Figure 7-4**.

Figure 7-4 Downloading an agent



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

NOTE

Agent running relies on Java 8. Before deploying an agent, ensure that Java 8 has been installed and Java environment variables have been configured.

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

sh sbin/install.sh

su Ruby

sh sbin/start.sh

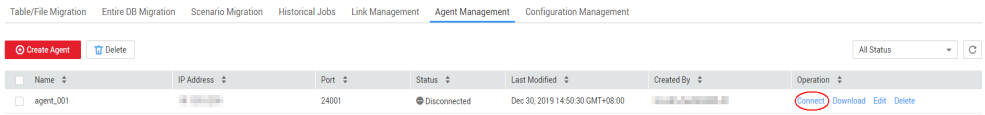
Step 4 After the installation is complete, run the **netstat -an** command to check whether listening is enabled for the port.

----End

Connecting to an Agent

Step 1 On the **Agent Management** page, locate the created agent. Click **Connect** in the **Operation** column. See **Figure 7-5**.

Figure 7-5 Connecting to an agent



Step 2 Select the agent in **Link to HDFS** and **Link to Relational Databases**.

----End

8 Migration Scenarios

8.1 Data Migration on the Cloud

8.1.1 From DDS to DWS

Scenario

CDM allows you to migrate data from DDS to other data sources. This section describes how to use CDM to migrate data from DDS to DWS. The procedure includes four steps:

1. **Creating a CDM Cluster and Binding an EIP to the Cluster**
2. **Creating a DDS Link**
3. **Creating a DWS Link**
4. **Creating a Migration Job**

Prerequisites

- You have purchased DWS and DDS.
- You have obtained the IP address, port number, database name, username, and password for connecting to the DWS and DDS databases. In addition, you must have the read, write, and delete permissions for the DDS and DWS databases.

Creating a CDM Cluster and Binding an EIP to the Cluster

Step 1 Log in to the **CDM management console** and create a CDM cluster. For details about how to create a CDM cluster, see **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.
- If DDS and DWS are deployed in the same VPC, the newly created CDM cluster also needs to be deployed in that VPC, with no EIP bound. The CDM

cluster's subnet and security group can be the same as those of the DDS or DWS cluster. You can also configure a security group rule to enable the CDM cluster to access the cluster of another service (DWS or DDS).

- If DDS and DWS are not deployed in the same VPC, the newly created CDM cluster needs to be in the same VPC as DDS and **an EIP must be bound** for the CDM cluster to access 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 DWS. If DDS and DWS are in the same VPC, do not bind an EIP to the CDM cluster.

NOTE

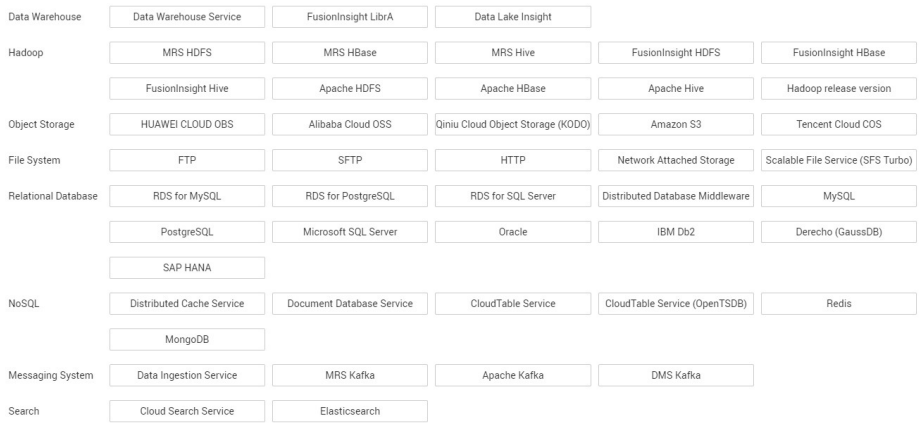
If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

Creating a DDS Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the page that is displayed, choose **Link Management** > **Create Link**. The page for selecting a connector is displayed. See **Figure 8-1**.

Figure 8-1 Selecting a connector



Step 2 To create a DDS link, select **Document Database Service** and click **Next**. On the page that is displayed, configure the link parameters based on **Table 8-1**.

Table 8-1 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	mongo_link

Parameter	Description	Example Value
Server List	Address list of the DDS cluster. The format is **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
Database Name	Name of the DDS database to be connected	DB_mongodb
Username	Username used for logging in to the DDS database	cdm
Password	Password used for logging in to the DDS database	-

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

----End

Creating a DWS Link

Step 1 On the **Link Management** tab page, click **Create Link** and select **Data Warehouse Service** to create a DWS link.

Step 2 Click **Next**. The page for configuring the DWS link parameters is displayed. Configure the mandatory parameters according to **Table 8-2** and retain the default values of the optional parameters.

Table 8-2 DWS link parameters

Parameter	Description	Example Value
Name	Unique link name	dwslink
Database Server	IP address or domain name of the DWS database server	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

Parameter	Description	Example Value
Agent	Click **Select** and select the agent created in **Connecting to an 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

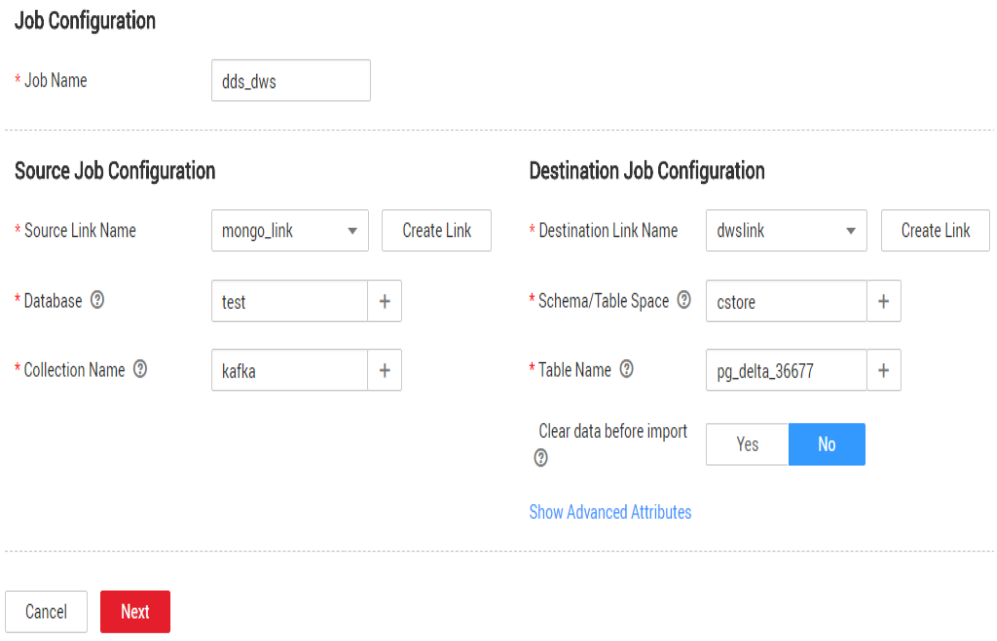
Step 3 Click **Save**.

----End

Creating a Migration Job

Step 1 Choose **Table/File Migration** > **Create Job** to create a data migration job.

Figure 8-2 Creating a job for migrating data from DDS to DWS



Step 2 Configure the required job information:

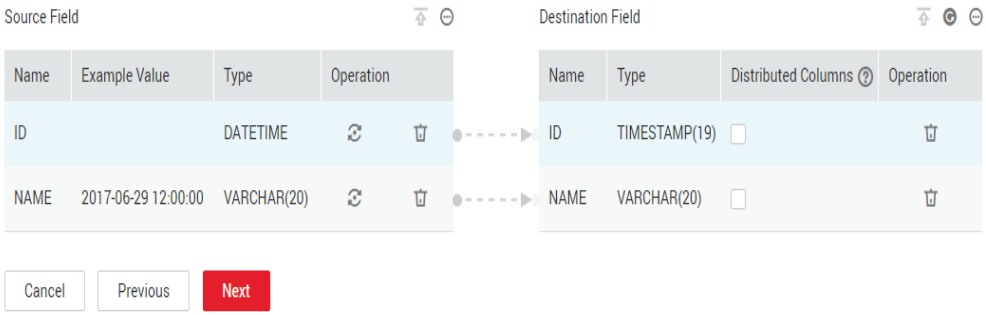
- **Job Name**: Enter a unique job name.
- **Source Job Configuration**
  - **Source Link Name**: Select the **mongo_link** link created in **Creating a DDS Link**.
  - **Database Name**: Select the database whose data is to be migrated.
  - **Collection Name**: Enter the name of the MongoDB collection on DDS, which is similar to the table name in a relational database.

- **Destination Job Configuration**
  - **Destination Link Name**: Select the **dwslink** link created in **Creating a DWS Link**.
  - **Schema/Tablespace**: Select the DWS database to which data is to be written.
  - **Table Name**: Name of the table to which data is to be written. You can manually enter a table name that does not exist. CDM automatically creates the table on DWS.
  - **Clear Data Before Import**: Choose whether to clear data in the destination table before data import.

Step 3 Click **Next**. The **Map Field** tab page is displayed. CDM automatically maps table fields at the migration source and destination. Check whether the field mapping is correct.

- If the field mapping is incorrect, click the row where the field is located and drag the field to adjust the mapping.
- When importing data to DWS, you need to manually select the distribution columns of DWS. 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 you need to convert the content of the source fields, perform the operations described in **Field Conversion**. In this example, the field conversion is not required.

Figure 8-3 Field mapping



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 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**: To configure scheduled jobs, see **Scheduling Job Execution**. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
- **Write Dirty Data**: Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS for future viewing. Before writing dirty data, create an OBS link. Retain the default value **No** so that dirty data is not recorded.
- **Delete Job After Completion**: Retain the default value **Do not delete**.

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

8.1.2 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

Log in to the **CDM management console** and create a CDM cluster. For details about how to create a CDM cluster, see **Creating a CDM Cluster**. The key configurations are as follows:

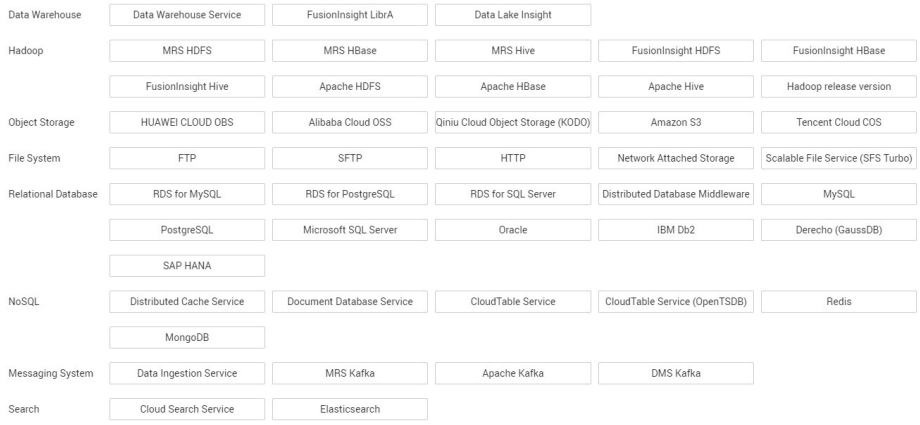
- The flavor of the CDM cluster is selected based on the amount of data to be migrated. Generally, cdm.medium meets the requirements for most migration scenarios.
- The CDM and Cloud Search Service clusters must be in the same VPC. In addition, it is recommended that the CDM cluster be in the same subnet 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 page that is displayed, choose **Link Management** > **Create Link**. The page for selecting a connector is displayed. See **Figure 8-4**.

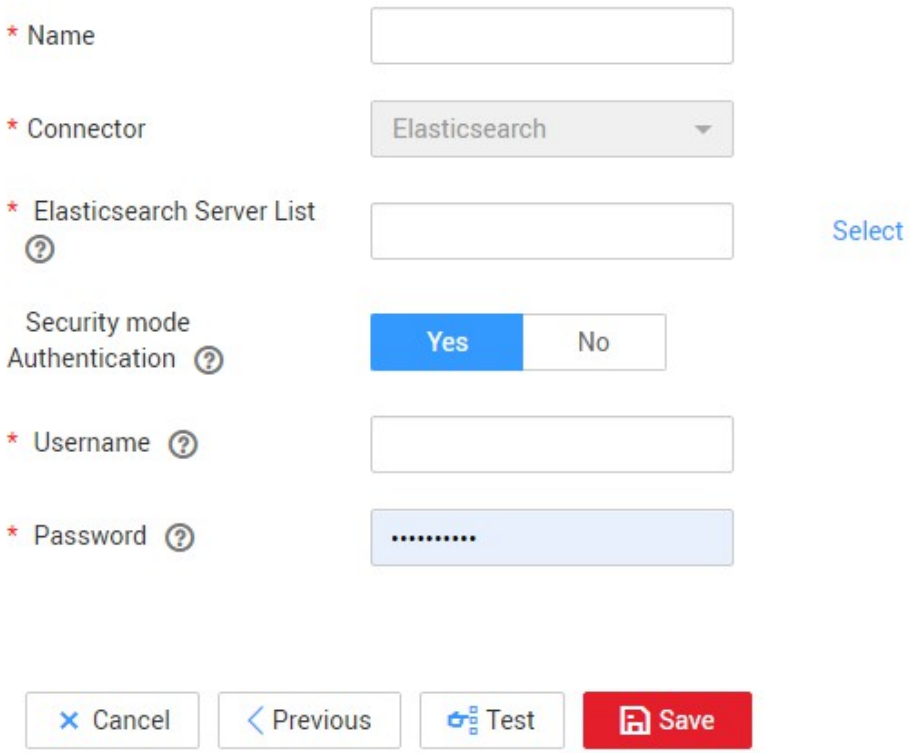
Figure 8-4 Selecting a connector



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 8-5 Creating a CSS link



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

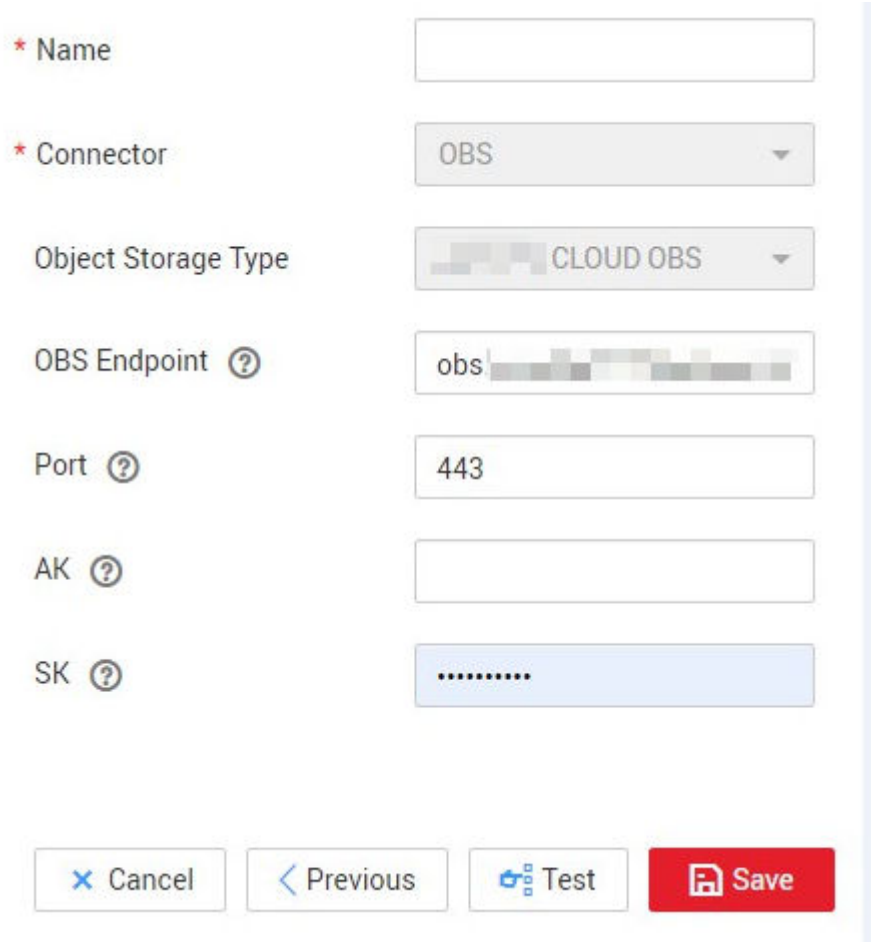
----End

Creating an OBS Link

Step 1 On the **Link Management** tab page, click **Create Link**. On the page displayed, select **HUAWEI CLOUD OBS**, click **Next**, and configure the required link parameters. See **Figure 8-6**.

- **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.

Figure 8-6 Creating an OBS link



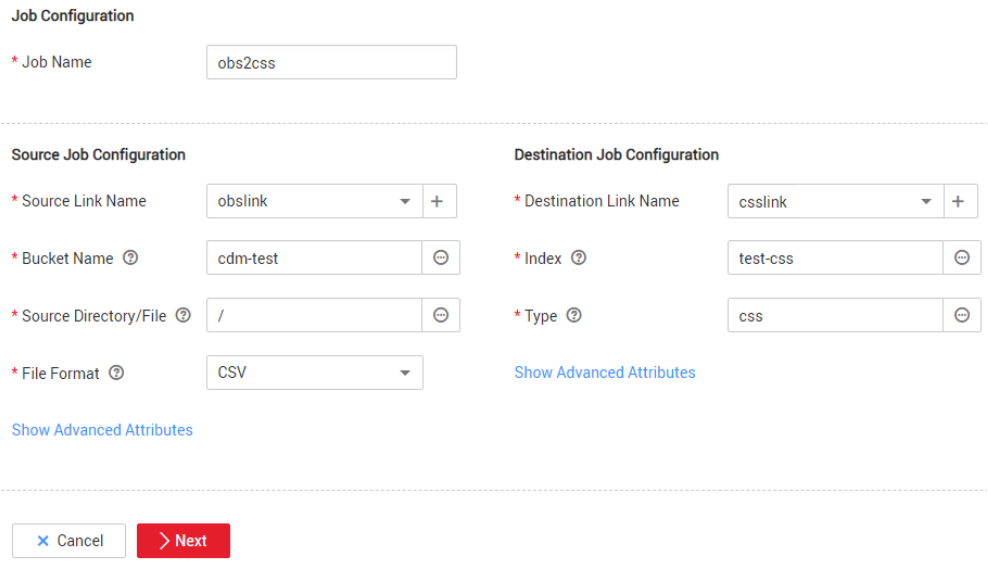
Step 2 Click **Save**. The **Link Management** page is displayed.

----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 8-7 Creating a job for migrating data from OBS to Cloud Search Service

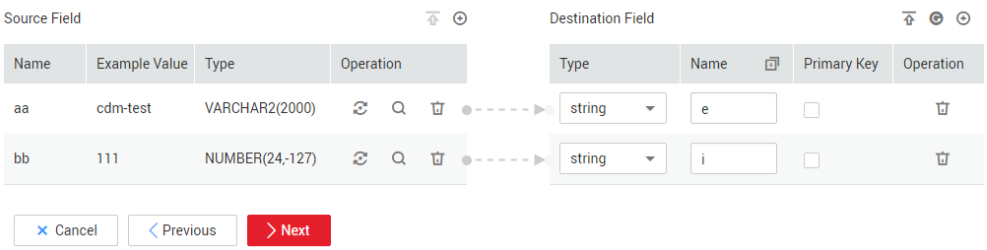


- **Job Name**: Enter a unique name.
- **Source Job Configuration**
  - **Source Link Name**: Select the **obslink** link created in **Creating an OBS Link**.
  - **Bucket Name**: Select the bucket from which the data will be migrated.
  - **Source Directory/File**: Set this parameter to the path of 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**. For details, see **From OBS/OSS/KODO/COS/S3**.
- **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**. For details, see **To Elasticsearch or CSS**.

Step 2 Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields. See **Figure 8-8**.

- 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 **Field Conversion**.

Figure 8-8 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 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**: To configure scheduled jobs, see **Scheduling Job Execution**. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
- **Write Dirty Data**: Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS for future viewing. Before writing dirty data, create an OBS link. Retain the default value **No** so that dirty data is not recorded.
- **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

8.1.3 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 subscribed to OBS and DLI.
- You have created resource queues, databases, and tables on DLI.

Creating a CDM Cluster

Log in to the **CDM management console**, and perform operations as required.

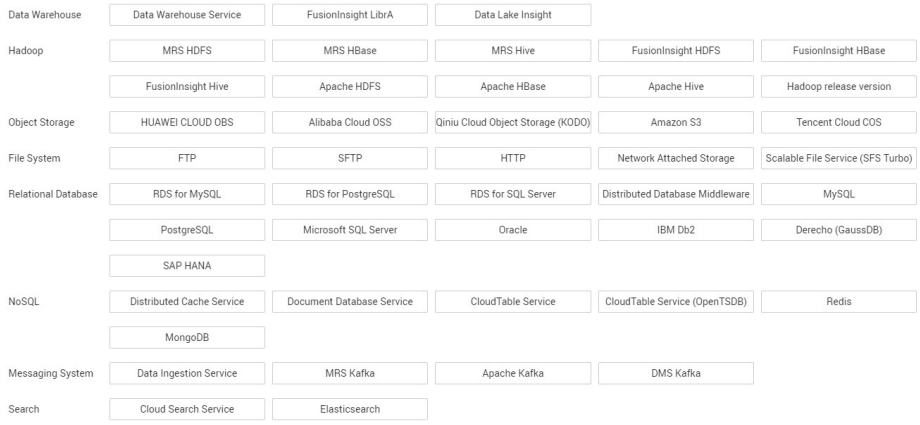
- If you already have a CDM cluster, click **Job Management** in the row of the cluster and create links on the page that is displayed.
- If you do not have a CDM cluster, click **Buy CDM Cluster** to create a cluster. For details about how to create a cluster, see **Creating a CDM Cluster**.

  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 page that is displayed, choose **Link Management** > **Create Link**. The page for selecting a connector is displayed. See **Figure 8-9**.

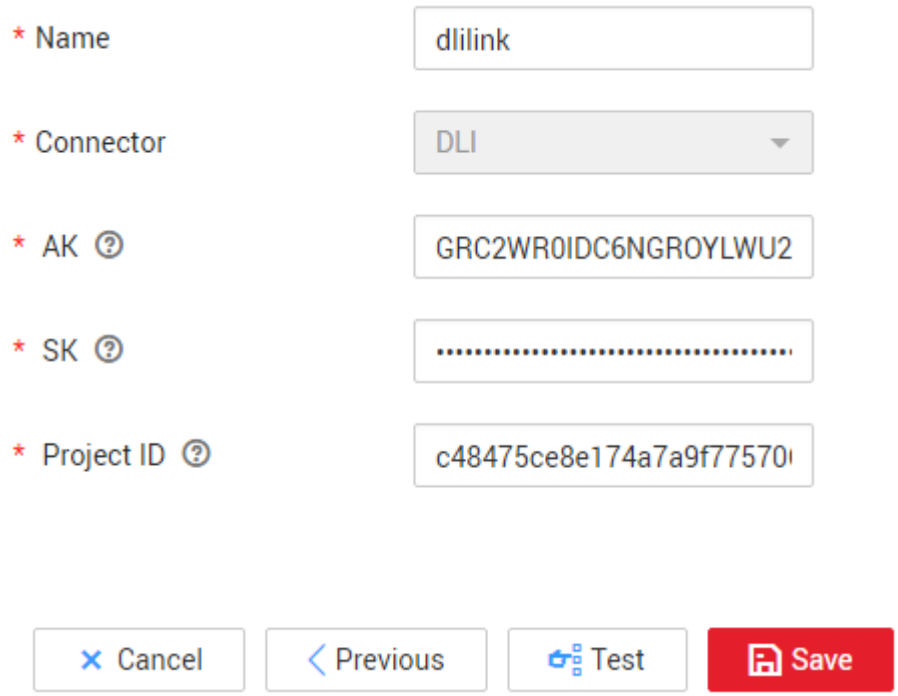
Figure 8-9 Selecting a connector



Step 2 Select **Data Lake Insight**, click **Next**, and configure the DLI link parameters. See **Figure 8-10**.

- **Name**: Enter a custom link name, for example, **dlilink**.
- **AK** and **SK**: Enter the AK and SK used for accessing the DLI database. To obtain the AK and SK, hover the cursor on the username on the management console and choose **My Credentials** > **Access Keys**.
- **Project ID**: Enter the ID of the project to which DLI belongs. Obtain the project ID on the **My Credentials** page.

Figure 8-10 Creating a DLI link



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

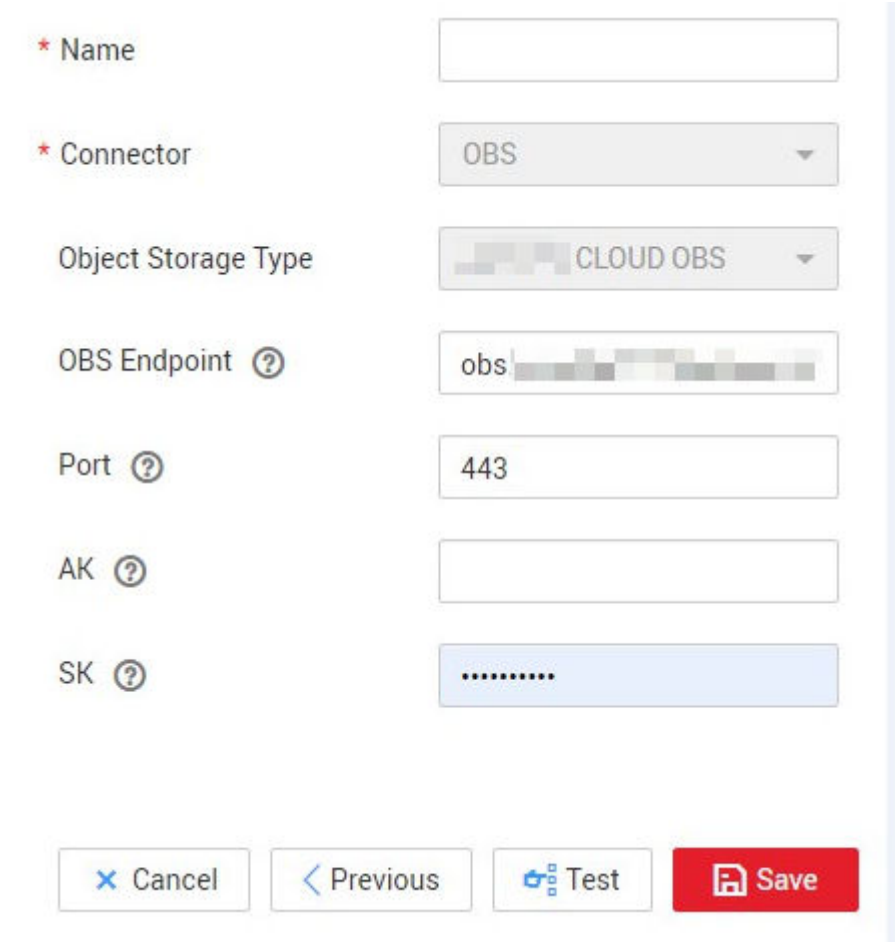
----End

Creating an OBS Link

Step 1 On the **Link Management** tab page, click **Create Link**. On the page displayed, select **HUAWEI CLOUD OBS**, click **Next**, and configure the required link parameters. See **Figure 8-11**.

- **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.

Figure 8-11 Creating an OBS link



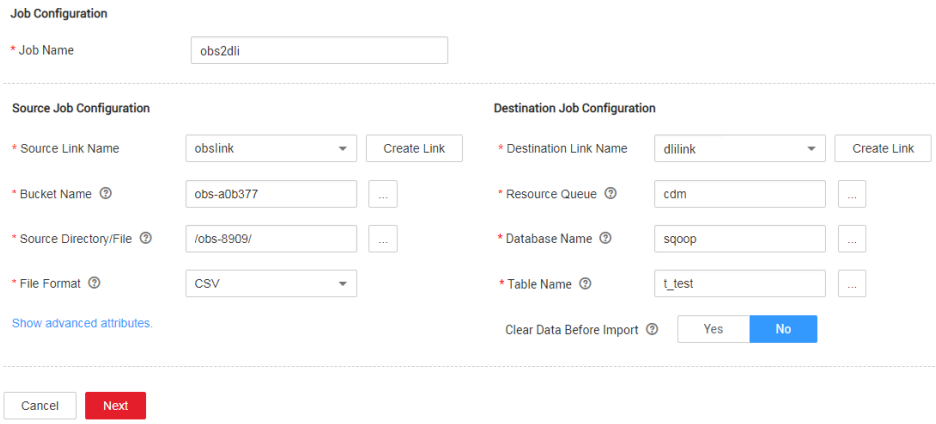
Step 2 Click **Save**. The **Link Management** page is displayed.

----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 8-12**.

Figure 8-12 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**. For details, see **From OBS/OSS/KODO/COS/S3**.
- **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 **Field Conversion**.

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**: To configure scheduled jobs, see **Scheduling Job Execution**. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
- **Write Dirty Data**: Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS for future viewing. Before writing dirty data, create an OBS link. Retain the default value **No** so that dirty data is not recorded.
- **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

8.2 Database Migration

8.2.1 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 sufficient EIP quota.
- 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.

Creating a CDM Cluster and Binding an EIP to the Cluster

Step 1 Log in to the **CDM management console** and create a CDM cluster. For details about how to create a CDM cluster, see **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

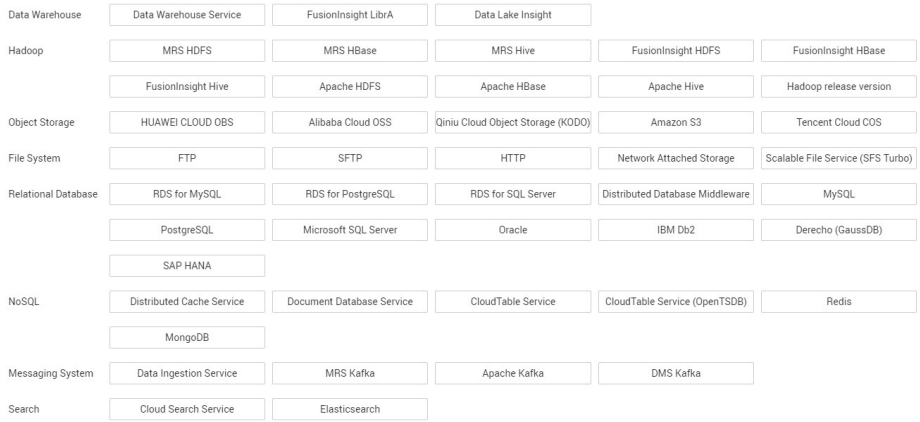
If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

Creating a Cloud Search Service Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the page that is displayed, choose **Link Management** > **Create Link**. The page for selecting a connector is displayed. See **Figure 8-13**.

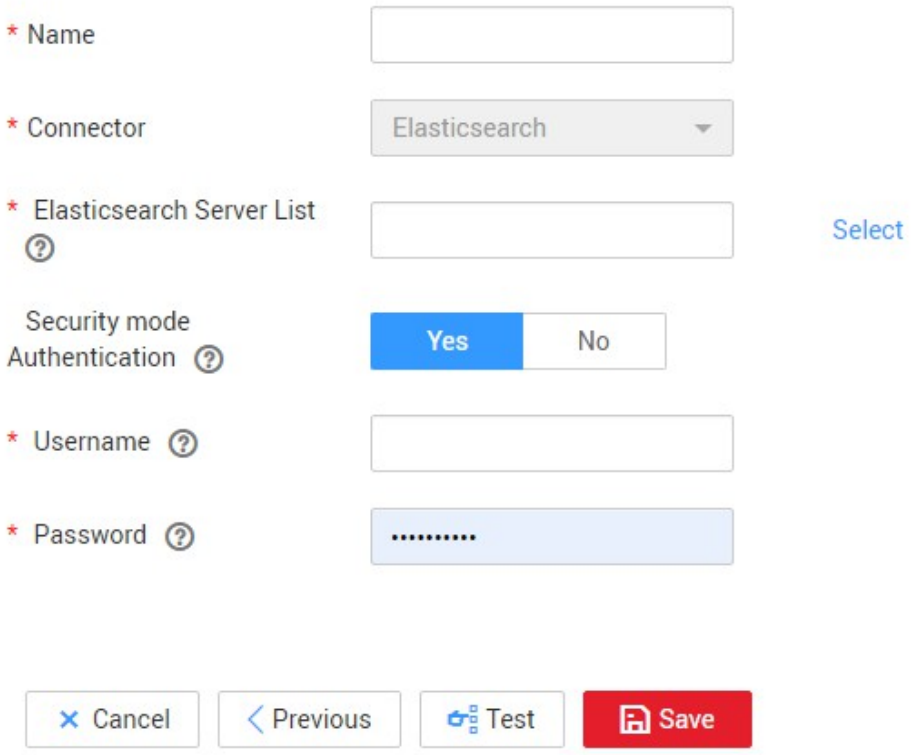
Figure 8-13 Selecting a connector



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 8-14 Creating a CSS link



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

----End

Creating an Oracle Link

Step 1 On the **Link Management** tab page, click **Create Link**. On the page that is displayed, select **Oracle**, click **Next**, and configure the Oracle link parameters.

- **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.

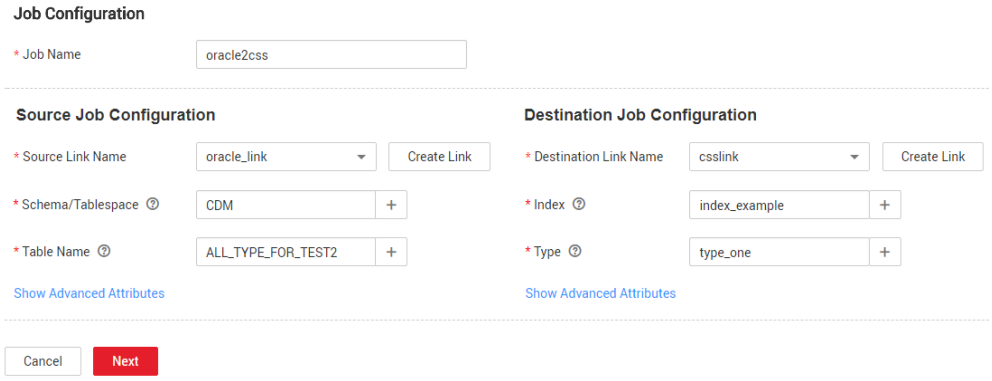
Step 2 Click **Save**. The **Link Management** page is displayed.

----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 8-15 Creating a job for migrating data from Oracle to Cloud Search Service

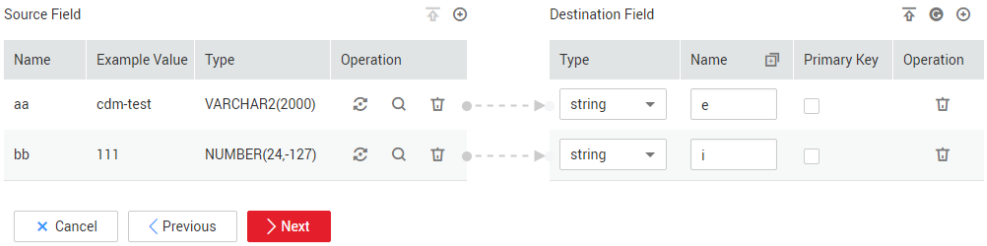


- **Job Name**: Enter a unique name.
- **Source Job Configuration**
  - **Source Link Name**: Select the **oracle_link** link created in **Creating an Oracle Link**.
  - **Schema/Tablespace**: Enter the name of the database whose data is to be migrated.
  - **Table Name**: Enter the name of the table to be migrated.
  - Retain the default values of the optional parameters in **Show Advanced Attributes**. For details, see **From a Relational Database**.
- **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**. For details, see **To Elasticsearch or CSS**.

Step 2 Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields. See **Figure 8-16**.

- 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 **Field Conversion**.

Figure 8-16 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 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**: To configure scheduled jobs, see **Scheduling Job Execution**. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
- **Write Dirty Data**: Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS for future viewing. Before writing dirty data, create an OBS link. Retain the default value **No** so that dirty data is not recorded.
- **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

8.2.2 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 8-17**.

Figure 8-17 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

- You have subscribed to MRS.
- You have sufficient EIP quota.
- You have obtained the IP address, port number, 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.

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, greatly improving the performance.

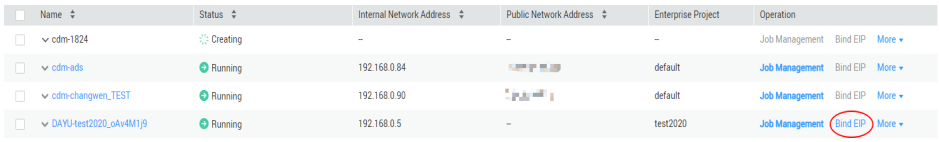
Creating a CDM Cluster and Binding an EIP to the Cluster

Step 1 Log in to the **CDM management console** and create a CDM cluster. For details about how to create a CDM cluster, see **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 8-18 Binding an EIP



NOTE

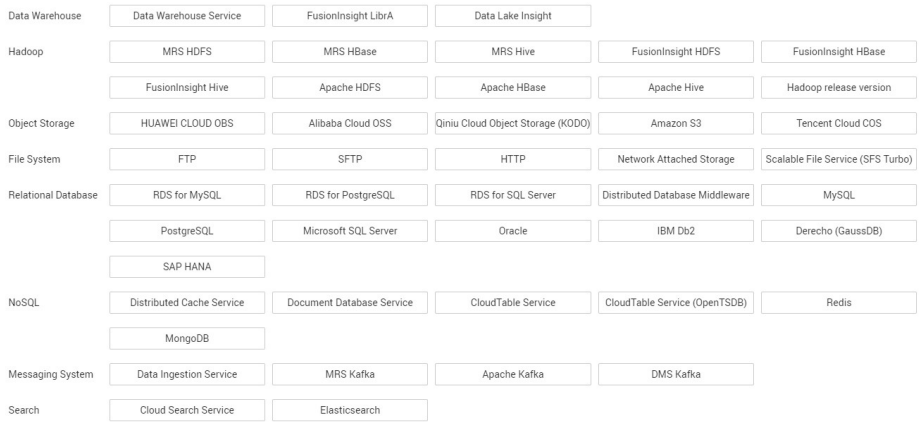
If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

Creating a MySQL Link

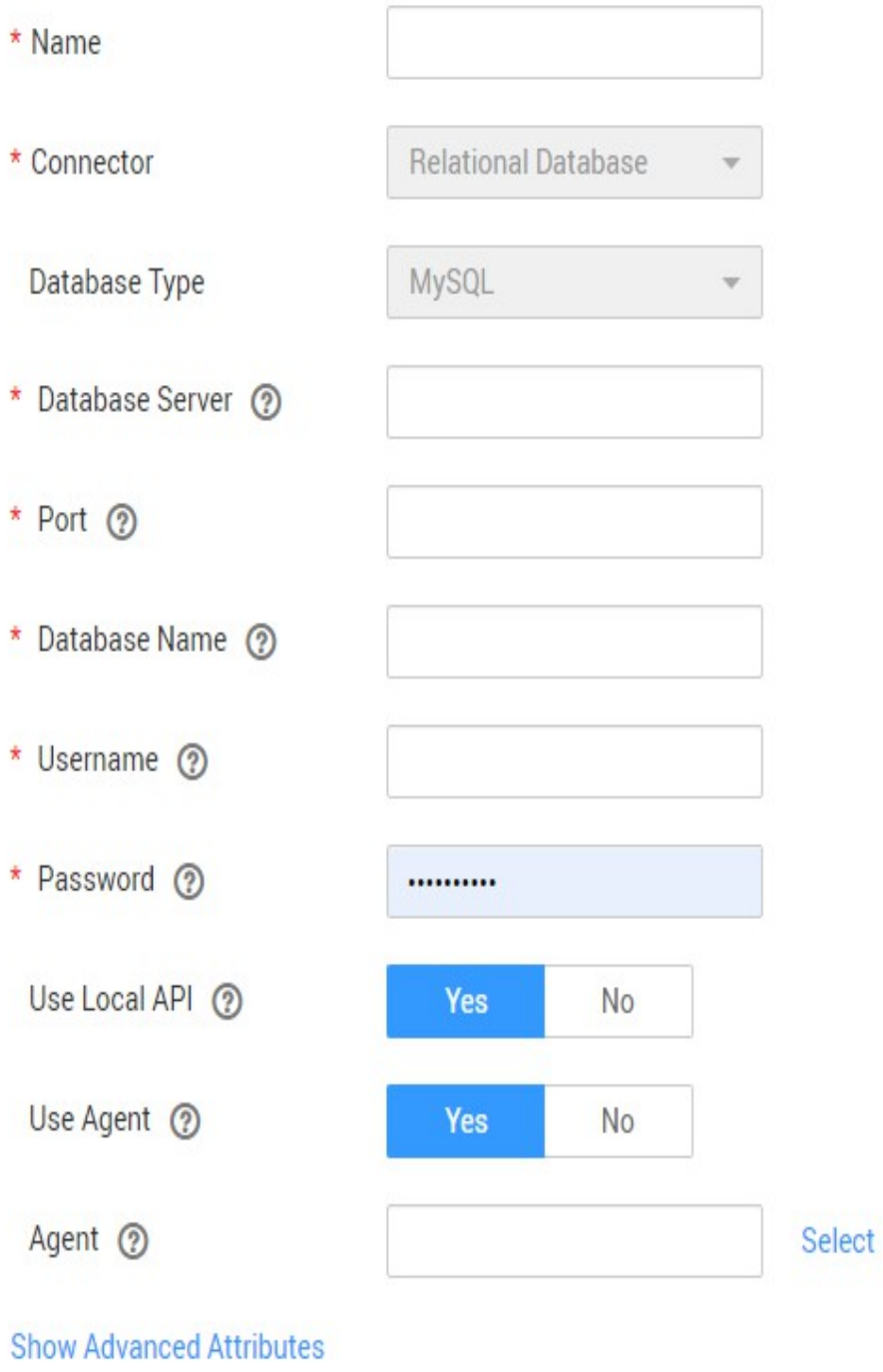
Step 1 On the **Cluster Management** page, click **Job Management** of the cluster and choose **Link Management** > **Create Link** to enter the page for selecting the connector. See **Figure 8-19**.

Figure 8-19 Selecting a connector



Step 2 Select **MySQL** and click **Next**. On the page that is displayed, configure MySQL link parameters.

Figure 8-20 Creating a MySQL link



Click **Show Advanced Attributes** to display optional parameters. For details, see **Link to Relational Databases**. 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

Parameter	Description	Example Value
Database Server	IP address or domain name of the MySQL database server	192.168.0.1
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 Agent	Whether to extract data from the data source through an agent	Yes
Agent	Click **Select** and select the agent created in **Connecting to an Agent**.	-

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 **Create Link** and select **MRS Hive** to create an MRS Hive link.

Step 2 Click **Next** and configure the MRS Hive link parameters. See **Figure 8-21**.

Figure 8-21 Creating a Hive link

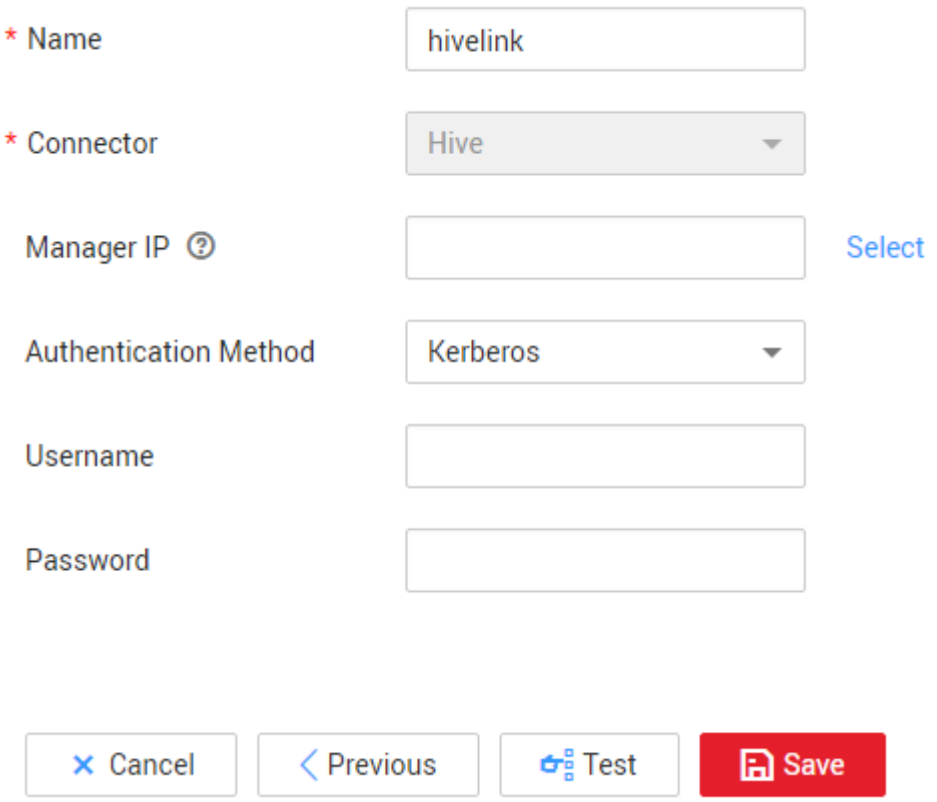


Table 8-4 describes the parameters. You can configure the parameters according to the actual situation.

Table 8-4 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	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
Manager Port	FusionInsight/MRS Manager port	28443
CAS Server Port	CAS protocol port of FusionInsight/MRS Manager	20009

Parameter	Description	Example Value
Authentication Method	Authentication method used for accessing MRS • **SIMPLE**: Select this if MRS is in non-security mode. • **KERBEROS**: Select this if MRS is in security mode.	SIMPLE
HIVE Version	Hive version	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.	cdm
Password	Password used for logging in to MRS Manager	-
OBS storage support	The server must support OBS storage. When creating a Hive table, you can store the table in OBS.	No
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, 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.	EMBEDDED

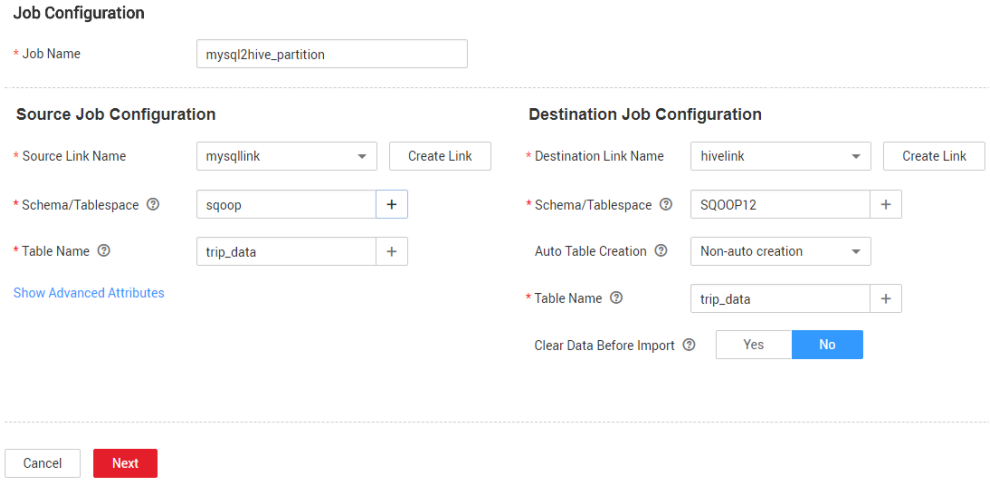
Step 3 Click **Save**. The **Link Management** page is displayed.

----End

Creating a Migration Job

Step 1 Choose **Table/File Migration** > **Create Job** to create a data migration job. **Figure 8-22** illustrates how to create a migration job.

Figure 8-22 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 8-23**.

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 8-23 Hive field mapping

Step 3 Click to display the **Converter List** dialog box, and then choose **Create Converter** > **Expression conversion**. See **Figure 8-24**.

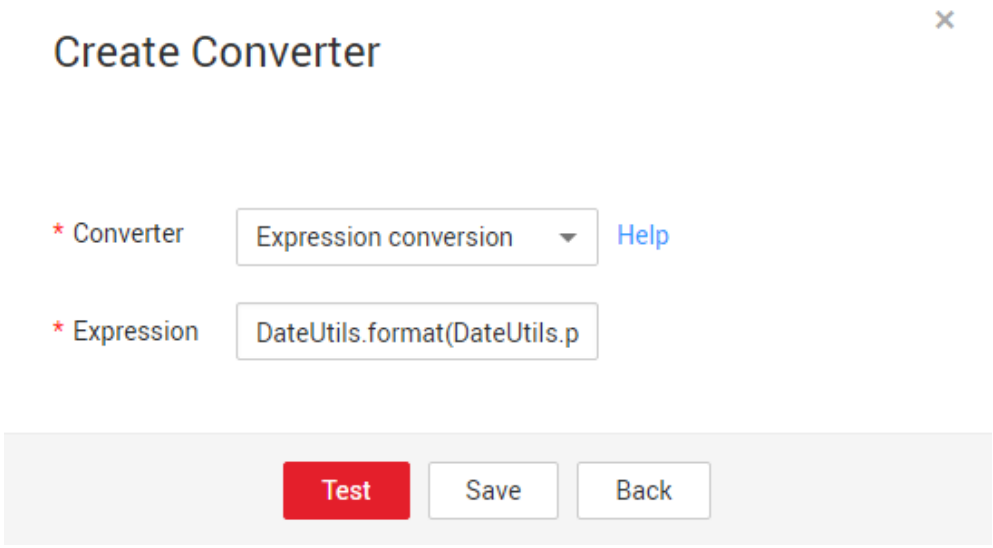
The expressions for the **y**, **ym**, and **ymd** fields are as follows:

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 8-24 Configuring the expression



NOTE

The expressions in CDM support field conversion of common character strings, dates, and values. For details, see **Field Conversion**.

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 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**: To configure scheduled jobs, see **Scheduling Job Execution**. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
- **Write Dirty Data**: Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS for future viewing. Before writing dirty data, create an OBS link. Retain the default value **No** so that dirty data is not recorded.
- **Delete Job After Completion**: Retain the default value **Do not delete**.

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

8.3 File Migration

8.3.1 From OSS to OBS

Scenario

CDM allows you to directly migrate object storage data from a third-party cloud to OBS without forwarding or writing code.

This section describes how to use CDM to migrate files from OSS on Alibaba Cloud to OBS. The procedure is as follows:

1. **Creating a CDM Cluster and Binding an EIP to the Cluster**
2. **Creating an OBS Link**
3. **Creating an OSS Link**
4. **Creating a Migration Job**

Preparing Data

- Endpoint for accessing OSS, for example, oss-cn-hangzhou.aliyuncs.com
- AK, temporary credential, or security token for accessing OSS
- Domain name, port number, AK, and SK for accessing OBS

Creating a CDM Cluster and Binding an EIP to the Cluster

Step 1 Log in to the **CDM management console** and create a CDM cluster. For details about how to create a CDM cluster, see **Creating a CDM Cluster**. The key configurations are as follows:

- Select the **cdm.medium** instance, which is applicable to most migration scenarios.
- If the cluster is used only to migrate data from third-party data sources to OBS on HUAWEI CLOUD, there is no special requirements on the VPC, subnet, and security group of the CDM cluster. You can specify them based on your needs.

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 accesses Alibaba Cloud OSS through the public network.

Because data is imported to HUAWEI CLOUD, 5 Mbit/s bandwidth for the EIP is enough.

NOTE

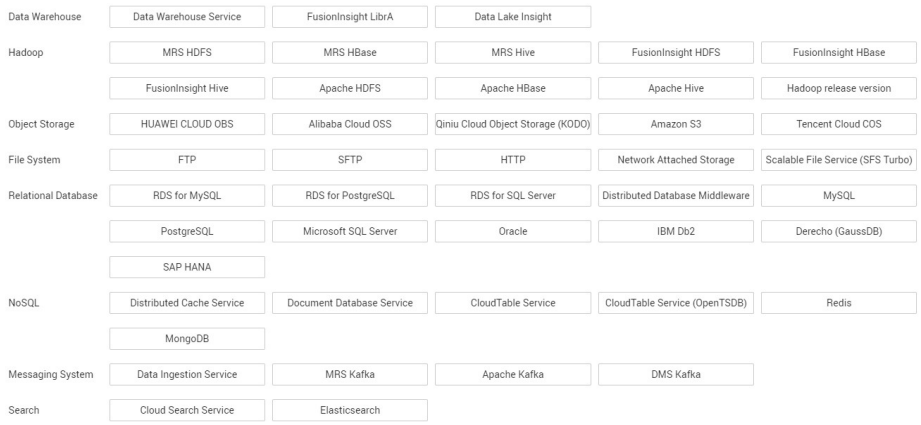
If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

Creating an OBS Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the page that is displayed, choose **Link Management** > **Create Link**. The page for selecting a connector is displayed. See **Figure 8-25**.

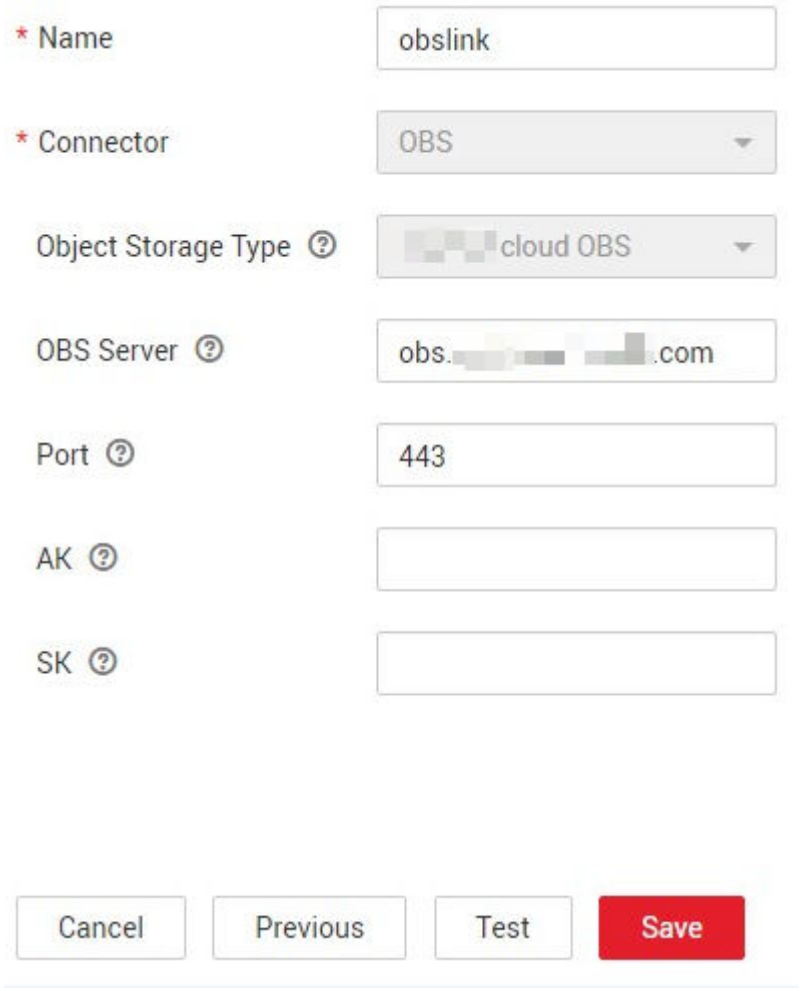
Figure 8-25 Selecting a connector



Step 2 Select **Object Storage Service** and click **Next**. On the page that is displayed, configure the OBS link parameters.

- **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.

Figure 8-26 Creating an OBS link



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

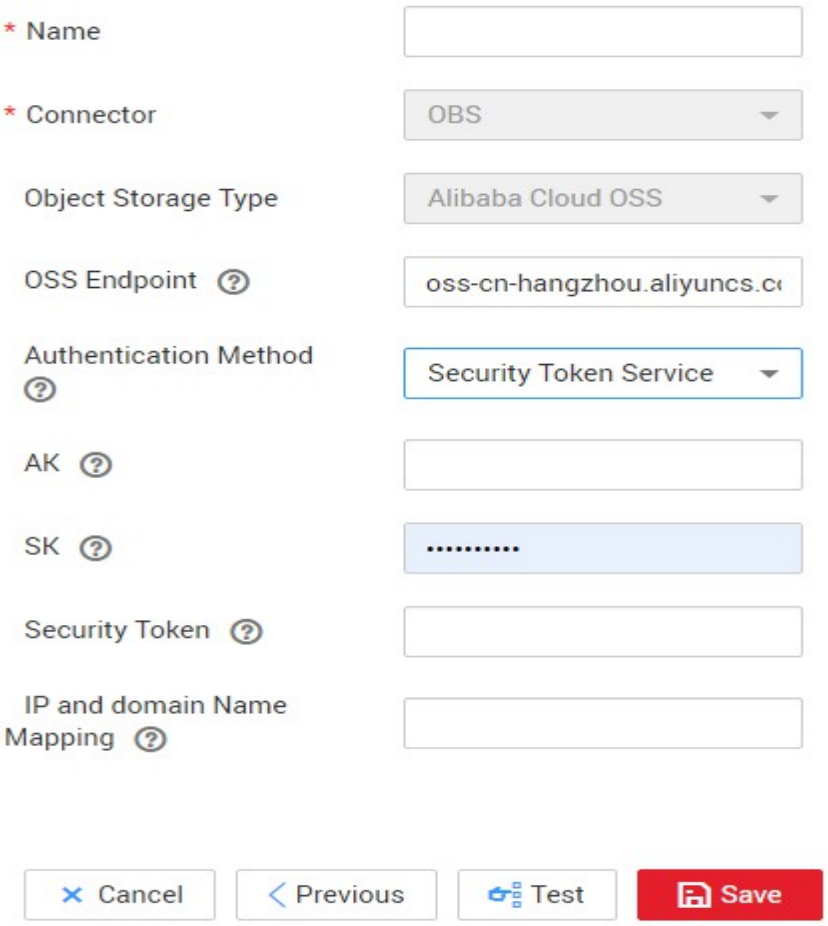
----End

Creating an OSS Link

Step 1 On the **Link Management** tab page, click **Create Link**. On the page that is displayed, select **Alibaba Cloud OSS**, click **Next**, and configure the required link parameters. See **Figure 8-27**.

- **Name**: Enter a custom link name.
- **OSS Endpoint**: Configure the endpoint of the data to be migrated. Select the endpoint based on the region where the bucket resides.
- **Authentication Method**: Select an authentication method based on your needs, for example, **Access key**.
- **AK** and **SK**: Enter the AK and SK used for logging in to OSS.
- **IP and domain Name Mapping**: Configure the mapping between IP addresses and domain names.

Figure 8-27 Creating an OSS link



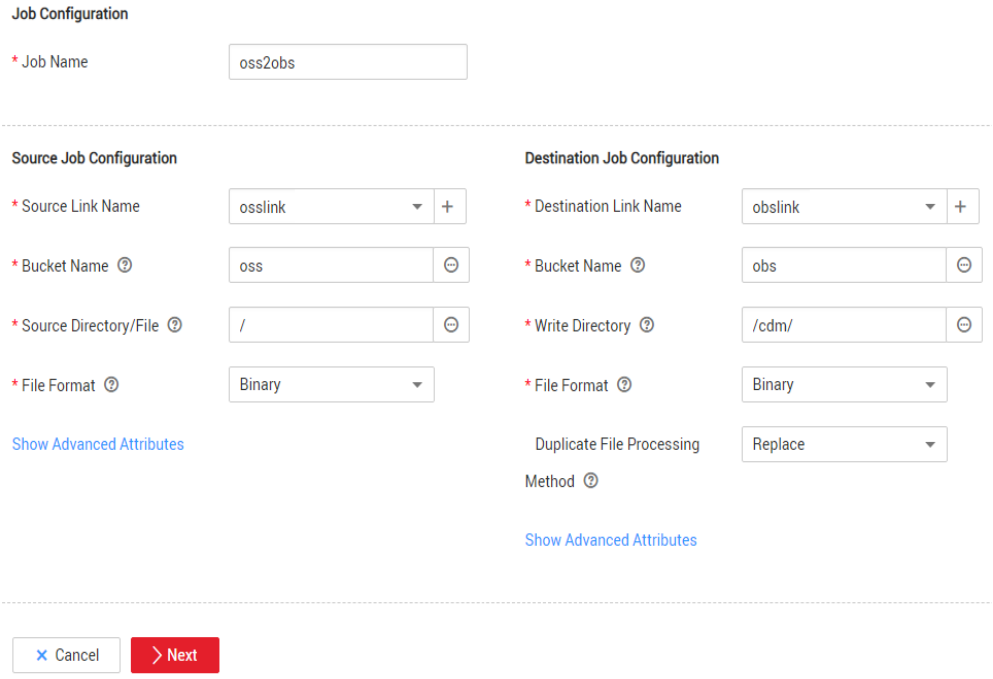
Step 2 Click **Save**. The **Link Management** page is displayed.

----End

Creating a Migration Job

Step 1 Choose **Table/File Migration** > **Create Job** to create a job for migrating data from OSS to OBS. See **Figure 8-28**.

Figure 8-28 Creating a job for migrating data from OSS to OBS



- **Job Name**: Enter a custom job name.
- **Source Job Configuration**
  - **Source Link Name**: Select the **osslink** link created in **Creating an OSS 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. You can migrate all files in the bucket.
  - **File Format**: Select **Binary**. It is applicable to file copy. To write files to databases, select **CSV** or **JSON**.
  - Retain the default values of the optional parameters in **Show Advanced Attributes**. For details, see **From OBS/OSS/KODO/COS/S3**.
- **Destination Job Configuration**
  - **Destination Link Name**: Select the **obslink** link created in **Creating an OBS Link**.
  - **Bucket Name**: Select the bucket to which data is to be written.
  - **Write Directory**: Select the path for storing data.
  - **File Format**: Select **Binary**. The value must be the same as that on the migration source.
  - Retain the default values of other optional parameters. For details, see **To OBS**.

Step 2 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**: To configure scheduled jobs, see **Scheduling Job Execution**. Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
- **Write Dirty Data**: Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS for future viewing. Before writing dirty data, create an OBS link. Retain the default value **No** so that dirty data is not recorded.
- **Delete Job After Completion**: Retain the default value **Do not delete**.

Step 3 Click **Save and Run**. The **Job Management** page is displayed, on which you can view the job execution progress and result.

Step 4 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 Incremental Migration

8.4.1 From FTP/SFTP to OBS

Scenario

CDM can periodically upload new files to OBS. You do not need to compile code or manually upload the files, but directly use the massive storage capabilities of OBS to back up files.

This section describes how to periodically back up FTP files to OBS with CDM.

For example, the **to_obs_test** directory on the FTP server contains one subdirectory **another_dir** and two files **file1** and **file2**. **file2** is in the **another_dir** directory. **Figure 8-29** shows the files. Configure a scheduled job of CDM to transfer these files to OBS and add **file3** and **file4** to the directory to verify that CDM can periodically transfer new files to OBS.

Figure 8-29 Files on the FTP server

```
to_obs_test/:
drwx------ 2 ftptest users 4096 Nov  9 11:54 another_dir
-rw------- 1 ftptest users 5933 Nov  9 11:50 file1.rar
to_obs_test/another_dir:
-rw------- 1 ftptest users 2199050 Nov  9 11:54 file2.zip
```


Prerequisites

- You have sufficient EIP quota.
- You have created an OBS bucket and obtained the access key (AK and SK).
- You have obtained the IP address, username, and password of the FTP server.
- If the FTP server is in the on-premises environment, ensure that the FTP server is accessible through the public network, 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 Log in to the **CDM management console** and click **Buy CDM Cluster** to create a CDM cluster. The key configurations are as follows:

- Select the **cdm.medium** instance, which is applicable to most migration scenarios.
- If the cluster is used only to migrate data from third-party data sources to OBS on HUAWEI CLOUD, there is no special requirements on the VPC, subnet, and security group of the CDM cluster. You can specify them based on your needs.

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 FTP server.

NOTE

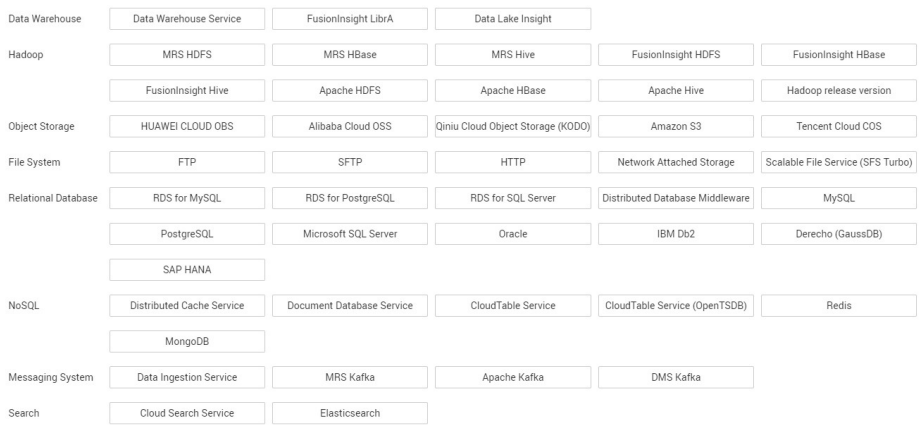
If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

Creating an OBS Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the page that is displayed, choose **Link Management** > **Create Link**. The page for selecting a connector is displayed. See **Figure 8-30**.

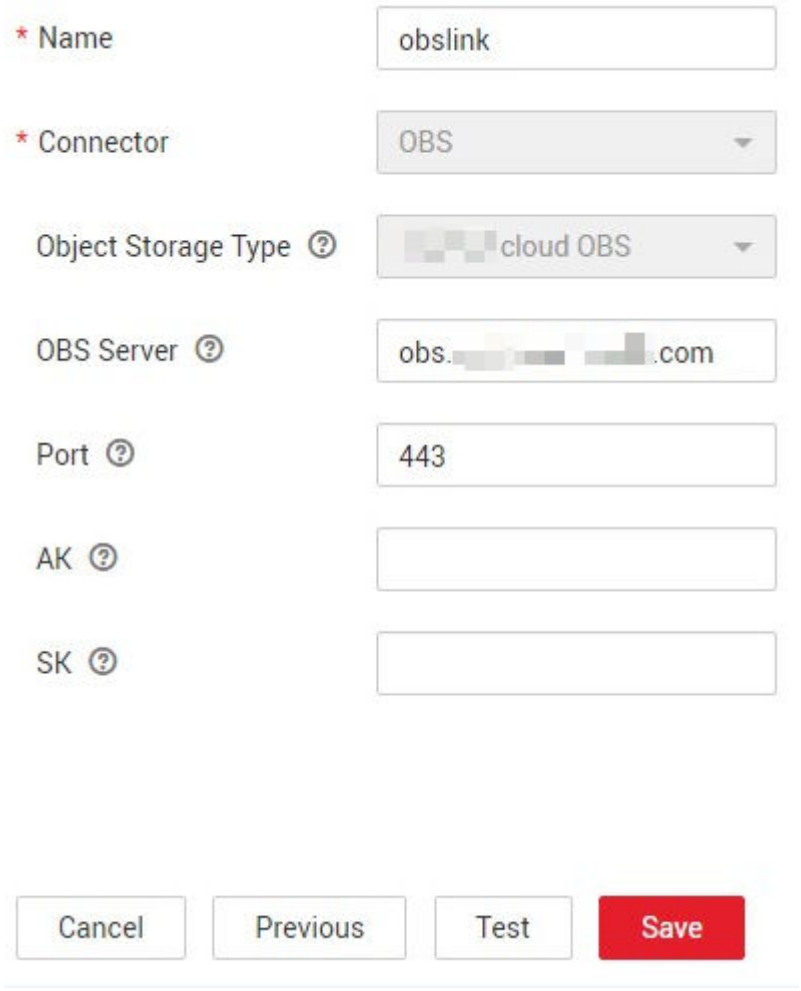
Figure 8-30 Selecting a connector



Step 2 Select **Object Storage Service** and click **Next**. On the page that is displayed, configure the OBS link parameters.

- **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.

Figure 8-31 Creating an OBS link



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

----End

Creating an FTP Link

Step 1 On the **Link Management** tab page, click **Create Link**. On the page that is displayed, select **FTP**, click **Next**, and configure the FTP link parameters.

- **Name**: Enter a custom link name, for example, **ftplink**.
- **Host Name/IP Address** and **Port**: Enter the address information about the FTP server.

- **Username** and **Password**: Enter the username and password used for logging in to the FTP server.

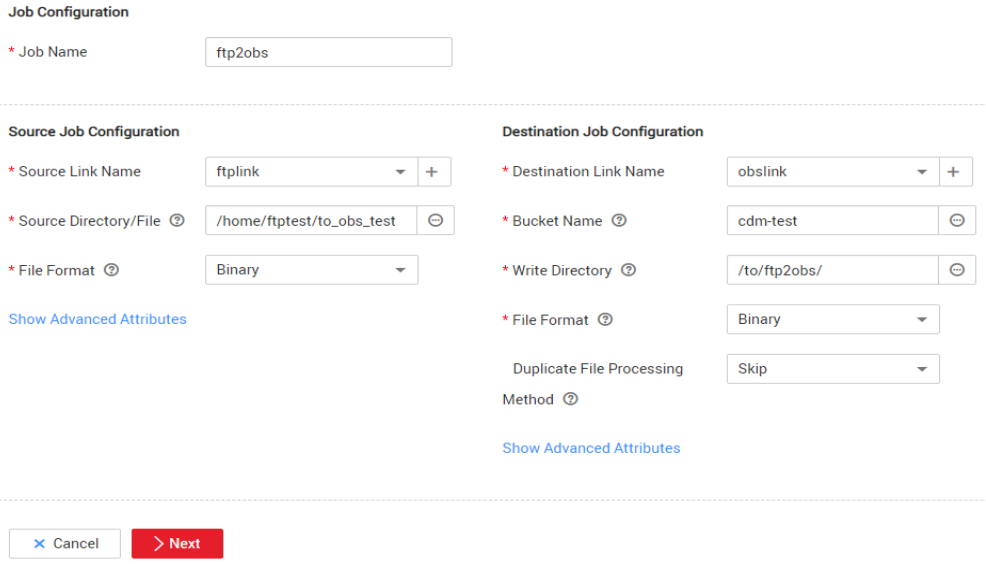
Step 2 Click **Save**. The **Link Management** page is displayed.

----End

Creating a Scheduled Migration Job

Step 1 Choose **Table/File Migration** > **Create Job** to create a data migration job.

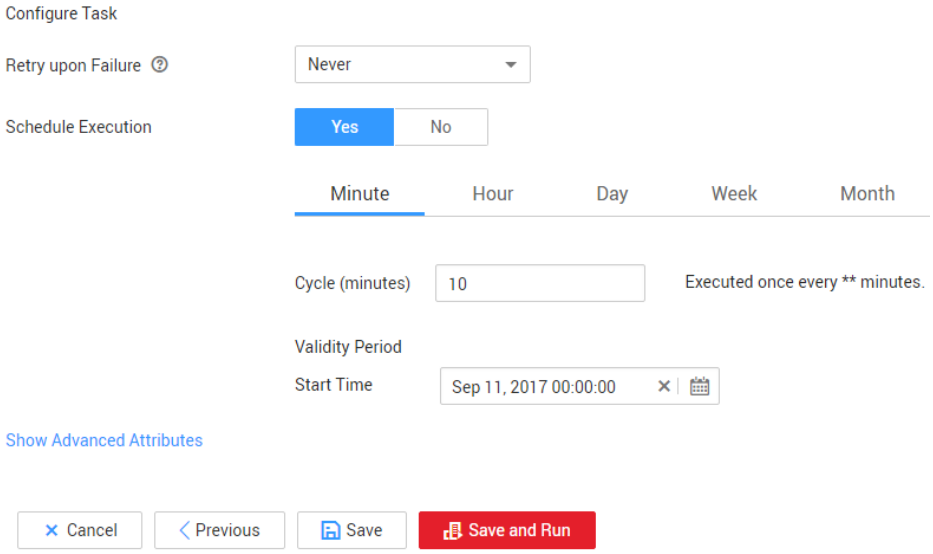
Figure 8-32 Creating a job for migrating data from FTP to OBS



- **Job Name**: Enter a custom job name.
- **Source Link Name**: Select the **ftplink** link created in **Creating an FTP Link**.
  - **Source Directory/File**: Select the path where **to_obs_test** is located.
  - **File Format**: Select **Binary**. It is applicable to file copy. To write files to databases, select **CSV** or **JSON**.
- **Destination Link Name**: Select the **obslink** link created in **Creating an OBS Link**.
  - **Bucket Name**: Select the OBS bucket for storing FTP files.
  - **Write Directory**: Select an existing directory or manually enter one. If the entered directory does not exist, CDM automatically creates one, for example, **/to/ftp2obs/**.
  - **File Format**: Select **Binary**. The value must be the same as that on the migration source.
  - **Duplicate File Processing Method**: Select **Skip** to avoid transferring duplicate files.

Step 2 Click **Next** and configure the scheduled task. In this example, the scheduled task is executed every 10 minutes. Retain the default values for other parameters.

Figure 8-33 Scheduling job execution



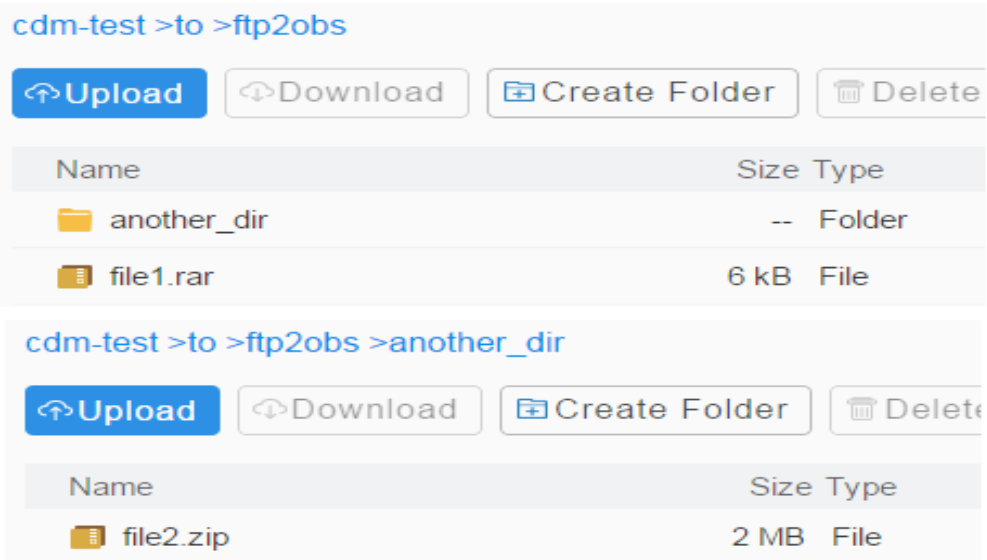
Step 3 Click **Save and Run**.

----End

Verifying Backup

Step 1 After the job is executed successfully, log in to the OBS client. You can see that the corresponding files exist on OBS. **Figure 8-34** shows the files on OBS.

Figure 8-34 Files on the OBS client



Step 2 In the FTP server directories, add files **file3** and **file4**. **file3** and **file1** are in the same directory, and **file2** and **file4** are in the same directory. See **Figure 8-35**.

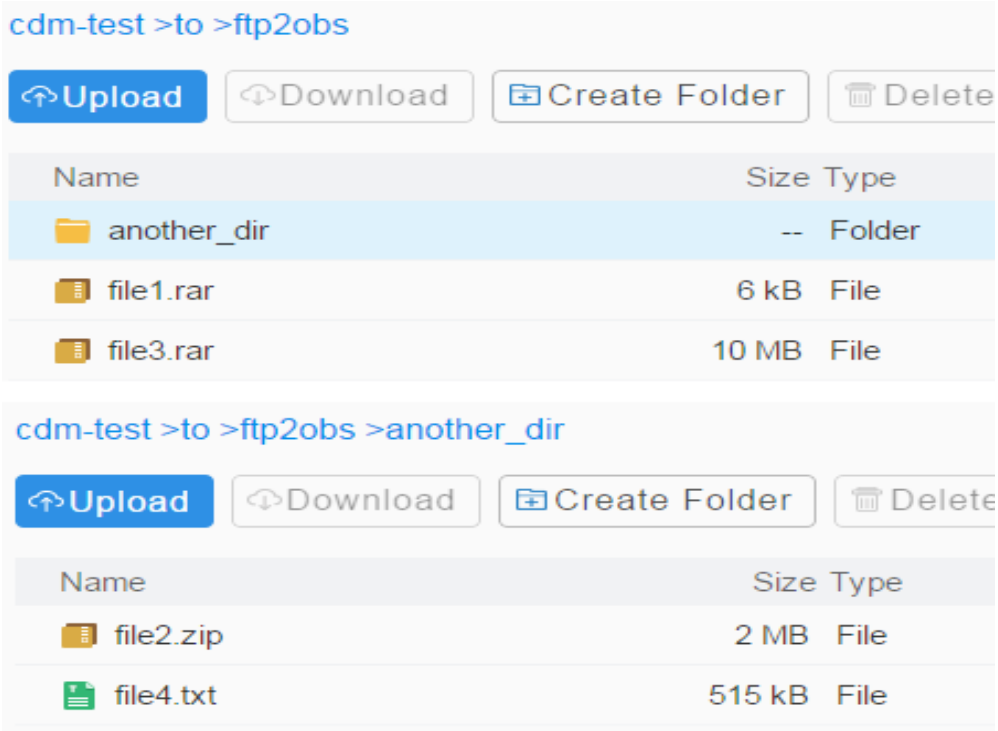
Figure 8-35 New files on the FTP server

```
to_obs_test/:
total 10356
drwx------ 2 ftptest users     4096 Nov  9 12:18 another_dir
-rw------- 1 ftptest users     5933 Nov  9 11:50 file1.rar
-rw------- 1 ftptest users 10575867 Nov  9 12:18 file3.rar

to_obs_test/another_dir:
total 2672
-rw------- 1 ftptest users 2199050 Nov  9 11:54 file2.zip
-rw------- 1 ftptest users  526921 Nov  9 12:18 file4.txt
```

Step 3 Wait 10 minutes and CDM automatically triggers the scheduled job. Then you can view the new files **file3** and **file4** after logging in to OBS. **Figure 8-36** shows the new files on OBS.

Figure 8-36 New files on OBS



Step 4 On the **Job Management** page, click **Historical Record** in the **Operation** column to view the job's historical execution records and read/write statistics.

Step 5 Click **Log** to view the job logs.

----End

8.4.2 Incremental Migration on CDM Supported by DLF

CDM supports incremental migration. For details, see **Incremental File Migration**, **Incremental Migration of Relational Databases**, **HBase/CloudTable Incremental Migration**, and **Incremental Synchronization Using the Macro Variables of Date and Time**.

Data Lake Factory (DLF) is a one-stop big data collaboration development platform. With DLF's online script editing, CDM migration jobs can be scheduled to implement incremental migration.

This section describes how DLF helps CDM migrate data from DWS to OBS. The procedure is as follows:

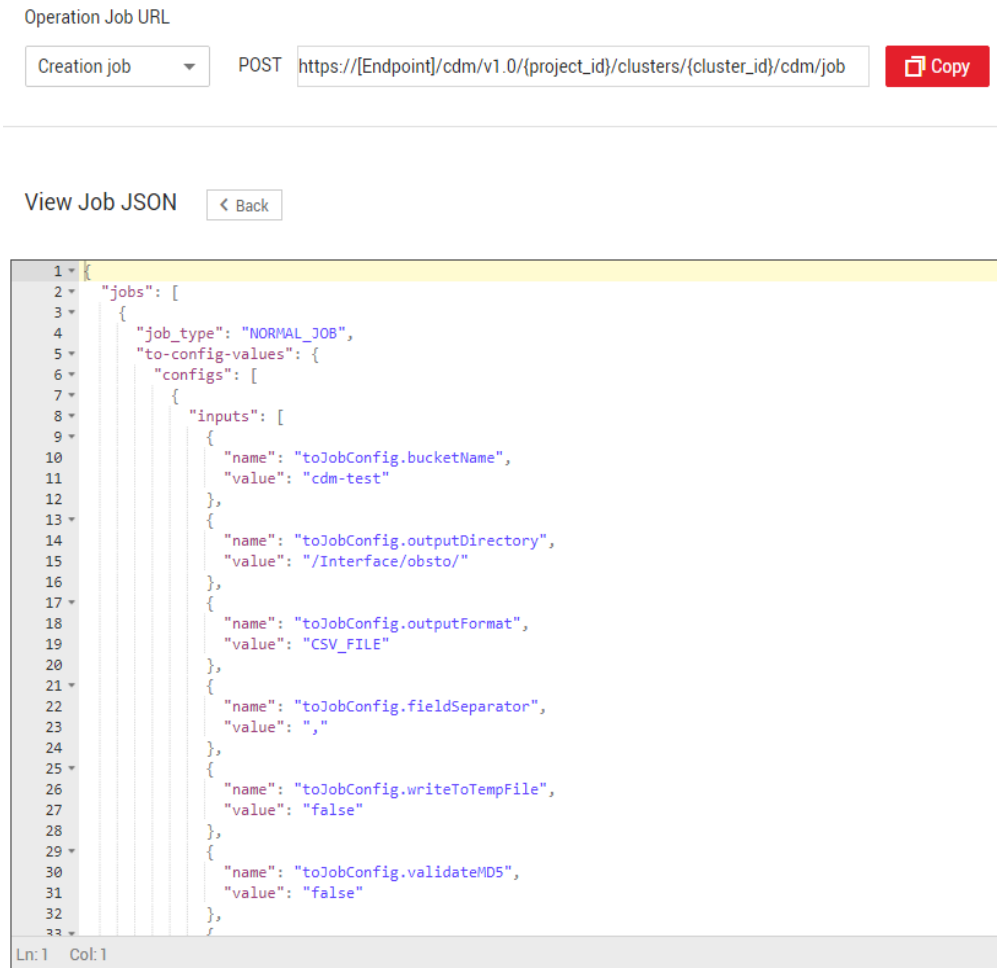
1. **Viewing a Job JSON File**
2. **Modifying a Job JSON File**
3. **Creating and Running a CDM Job on DLF**
4. **Waiting Until the Job Execution Is Completed**
5. **(Optional) Deleting a Job**
6. **Configuring DLF Job Parameters**

Viewing a Job JSON File

1. Log in to the **CDM management console** and create a table/file migration job for migrating data from DWS to OBS.
2. On the **Table/File Migration** tab page of the **Job Management** page, find the job you have created, and choose **More** > **View Job JSON** in the **Operation** column to view the job JSON file. See **Figure 8-37**.

   You can also use other job JSON files.

Figure 8-37 Viewing a job JSON file



3. A job JSON file is the request message body template for creating a CDM job. In the URL, **[Endpoint]**, **{project_id}**, and **{cluster_id}** must be replaced with the actual information.
   - **[Endpoint]**: Obtain the value from **Regions and Endpoints**, for example, **cdm.cn-north-1.myhuaweicloud.com**.
   - **{project_id}**: Project ID. Obtain the value from the project list on the **My Credentials** page
   - **{cluster_id}**: Cluster ID. Click the cluster name on the **Cluster Management** page to view the cluster ID.

Modifying a Job JSON File

You can modify the JSON body as required. The following uses 1 day as the cycle. The WHERE clause is used as the judgment condition for data extraction (generally, the time field is used as the judgment condition for incremental migration), and the data generated on the previous day is migrated every day.

1. Modify the WHERE clause to migrate incremental data in a certain period.

```
{
  "name": "fromJobConfig.whereClause",
  "value": "_timestamp >= '${startTime}' and _timestamp < '${currentTime}'"
}
```


NOTE

- If the migration source is DWS or a MySQL database, the time can be determined as follows:
  ```
  _timestamp >= '2018-10-10 00:00:00' and _timestamp < '2018-10-11 00:00:00'
  Or
  _timestamp between '2018-10-10 00:00:00' and '2018-10-11 00:00:00'
  ```
- If the migration source is an Oracle database, the WHERE clause is as follows:
  ```
  _timestamp >= to_date (2018-10-10 00:00:00' , 'yyyy-mm-dd hh24:mi:ss' ) and _timestamp < to_date (2018-10-10 00:00:00' , 'yyyy-mm-dd hh24:mi:ss' )
  ```

2. Incremental data in each period is imported to different directories.

   ```
   {
     "name": "toJobConfig.outputDirectory",
     "value": "dws2obs/${currentTime}"
   }
   ```

3. Dynamically generate the job name. Otherwise, the job cannot be created because the job name is duplicate.

   ```
   "to-connector-name": "obs-connector",
   "from-link-name": "dws_link",
   "name": "dws2obs-${currentTime}"
   ```

For details about how to modify more parameters, see the **Cloud Data Migration API Reference**. The modified JSON example is as follows:

```
{
  "jobs": [
    {
      "job_type": "NORMAL_JOB",
      "to-config-values": {
        "configs": [
          {
            "inputs": [
              {
                "name": "toJobConfig.bucketName",
                "value": "cdm-test"
              },
              {
                "name": "toJobConfig.outputDirectory",
                "value": "dws2obs/${currentTime}"
              },
              {
                "name": "toJobConfig.outputFormat",
                "value": "CSV_FILE"
              },
              {
                "name": "toJobConfig.fieldSeparator",
                "value": ","
              },
              {
                "name": "toJobConfig.writeToTempFile",
                "value": "false"
              },
              {
                "name": "toJobConfig.validateMD5",
                "value": "false"
              },
              {
                "name": "toJobConfig.encodeType",
                "value": "UTF-8"
              },
              {
                "name": "toJobConfig.duplicateFileOpType",
                "value": "REPLACE"
              },
              {
                "name": "toJobConfig.kmsEncryption",
                "value": "false"
```



```
          }
         ],
         "name": "toJobConfig"
        }
       ]
      },
      "from-config-values": {
       "configs": [
        {
         "inputs": [
          {
           "name": "fromJobConfig.schemaName",
           "value": "dws_database"
          },
          {
           "name": "fromJobConfig.tableName",
           "value": "dws_from"
          },
          {
           "name": "fromJobConfig.whereClause",
           "value": "_timestamp >= '${startTime}' and _timestamp < '${currentTime}'"
          },
          {
           "name": "fromJobConfig.columnList",
           "value":
"_tiny&_small&_int&_integer&_bigint&_float&_double&_date&_timestamp&_char&_varchar&_text"
          }
         ],
         "name": "fromJobConfig"
        }
       ]
      },
      "from-connector-name": "generic-jdbc-connector",
      "to-link-name": "obs_link",
      "driver-config-values": {
       "configs": [
        {
         "inputs": [
          {
           "name": "throttlingConfig.numExtractors",
           "value": "1"
          },
          {
           "name": "throttlingConfig.submitToCluster",
           "value": "false"
          },
          {
           "name": "throttlingConfig.numLoaders",
           "value": "1"
          },
          {
           "name": "throttlingConfig.recordDirtyData",
           "value": "false"
          },
          {
           "name": "throttlingConfig.writeToLink",
           "value": "obs_link"
          }
         ],
         "name": "throttlingConfig"
        },
        {
         "inputs": [],
         "name": "jarConfig"
        },
        {
         "inputs": [],
         "name": "schedulerConfig"
        },
```

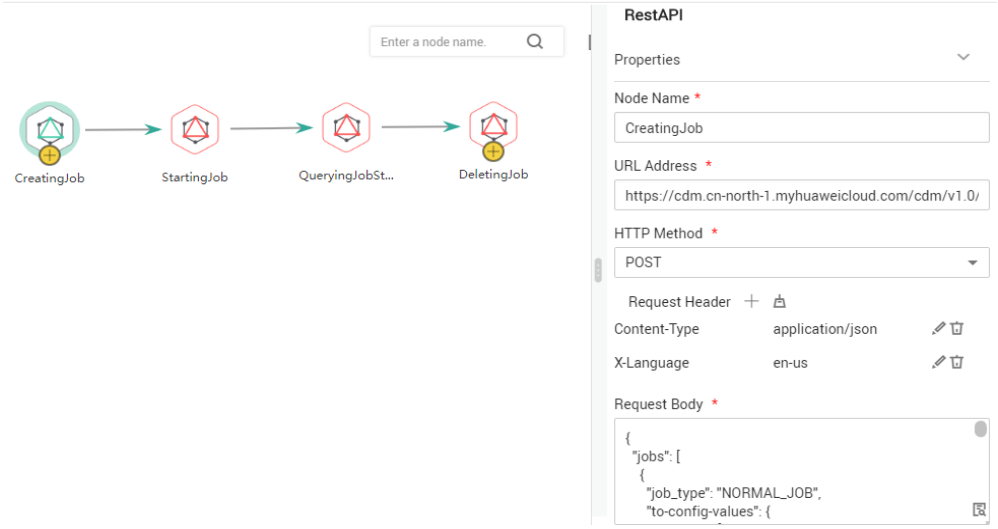


```
        {
          "inputs": [],
          "name": "transformConfig"
        },
        {
          "inputs": [],
          "name": "smnConfig"
        },
        {
          "inputs": [],
          "name": "retryJobConfig"
        }
      ]
    },
    "to-connector-name": "obs-connector",
    "from-link-name": "dws_link",
    "name": "dws2obs-${currentTime}"
  }
 ]
}
```

Creating and Running a CDM Job on DLF

1. For details about how to create a DLF job, see **Creating a Job** in the *Data Lake Factory User Guide*.
2. After the DLF job is created, double-click the job name to go to the job development page. DLF uses the RestAPI node to call a RESTful API to create a CDM migration job.
3. Configure the properties of the RestAPI node.
   a. **Node Name**: Customize a name, for example, **CreatingJob**. Note that the CDM migration job is only used as a node in the DLF job.
   b. **URL Address**: Enter the URL obtained in **Viewing a Job JSON File**, for example, **https://cdm.cn-north-1.myhuaweicloud.com/cdm/v1.0/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/job**.
   c. **HTTP Method**: Enter **POST**.
   d. Add the following request headers:
      - Content-Type = application/json
      - X-Language = en-us
   e. **Request Body**: Enter the modified JSON of the CDM job in **Modifying a Job JSON File**.

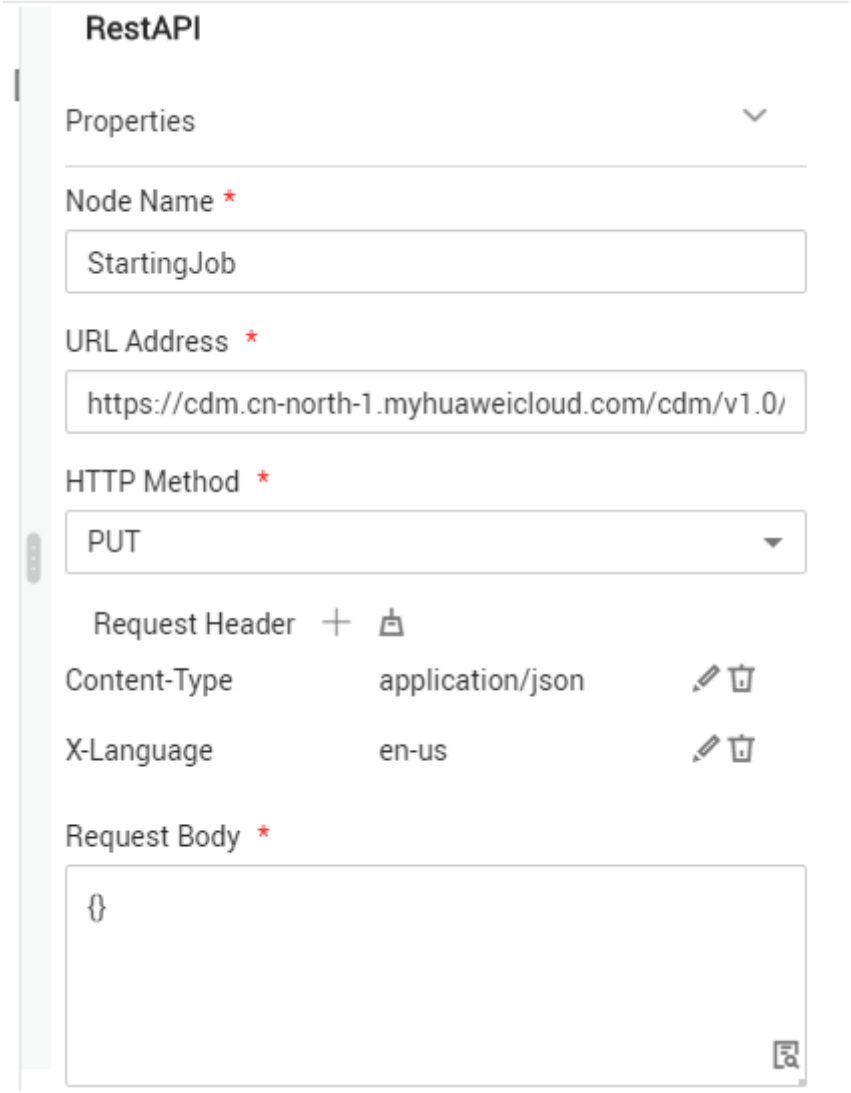
Figure 8-38 Properties of the node for creating the CDM job



4. After configuring the RestAPI node for creating a CDM job, you need to add the RestAPI node for running the CDM job. For details, see **Starting a Job** in the *Cloud Data Migration API Reference*.
   - **Node Name**: Customize a name, for example, **StartingJob**.
   - **URL Address**: Keep the values of **project_id** and **cluster_id** consistent with those in **the RestAPI node for creating the CDM job**. Set the job name to **dws2obs-${currentTime}**.

     For example, **https://cdm.cn-north-1.myhuaweicloud.com/cdm/v1.0/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/job/dws2obs-${currentTime}/start**.
   - **HTTP Method**: Enter **PUT**.
   - Add the following request headers:
     - Content-Type = application/json
     - X-Language = en-us

Figure 8-39 Properties of the node for running the CDM job



Waiting Until the Job Execution Is Completed

Because the CDM job is running asynchronously, the REST request for running the job returns **200**, which does not indicate that the data has been migrated successfully. If a computing job depends on the CDM migration job, a RestAPI node is required to periodically check whether the migration is successful. Computing is performed only when the migration is successful.

1. For details about the API for querying the migration job status, see **Querying Job Status** in the *Cloud Data Migration API Reference*.
2. After configuring the RestAPI node for running the CDM job, add the node for waiting for the CDM job completion. The node properties are as follows:
   - **Node Name**: Customize a name, for example, **WaitingJobCompletion**.
   - **URL Address**: For example, **https://cdm.cn-north-1.myhuaweicloud.com/cdm/v1.0/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/job/dws2obs-${currentTime}/status**.

  - **HTTP Method**: Enter **GET**.
  - Add the following request headers:
    - Content-Type = application/json
    - X-Language = en-us
  - Check Return Value: Select **YES**.
  - Property Path: Enter **submissions[0].status**.
  - **Request Success Flag**: Select **SUCCEEDED**.
  - Retain the default values of other parameters.

(Optional) Deleting a Job

If computing operations need to be performed after the migration, add various computing nodes to complete data computing.

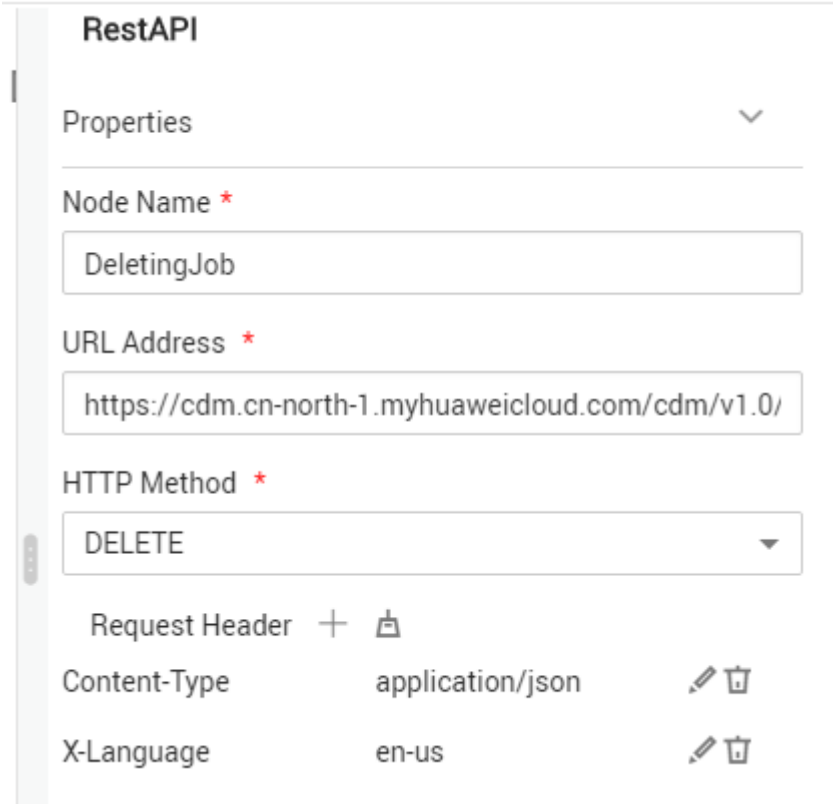
You can delete the CDM jobs as required. DLF periodically creates CDM jobs to implement incremental migration. Therefore, a large number of CDM jobs are accumulated on the CDM cluster. Therefore, after the migration is successful, you can delete the jobs that have been successfully executed.

If you need to delete a CDM job, add the RestAPI node for deleting the CDM job. Then, DLF calls the API in **Deleting a Job** in the *Cloud Data Migration API Reference*.

The node properties are as follows:

- **Node Name**: Customize a name, for example, **DeletingJob**.
- **URL Address**: For example, **https://cdm.cn-north-1.myhuaweicloud.com/cdm/v1.0/1551c7f6c808414d8e9f3c514a170f2e/clusters/6ec9a0a4-76be-4262-8697-e7af1fac7920/cdm/job/dws2obs-${currentTime}**.
- **HTTP Method**: Enter **DELETE**.
- Add the following request headers:
  - Content-Type = application/json
  - X-Language = en-us
- Retain the default values of other parameters.

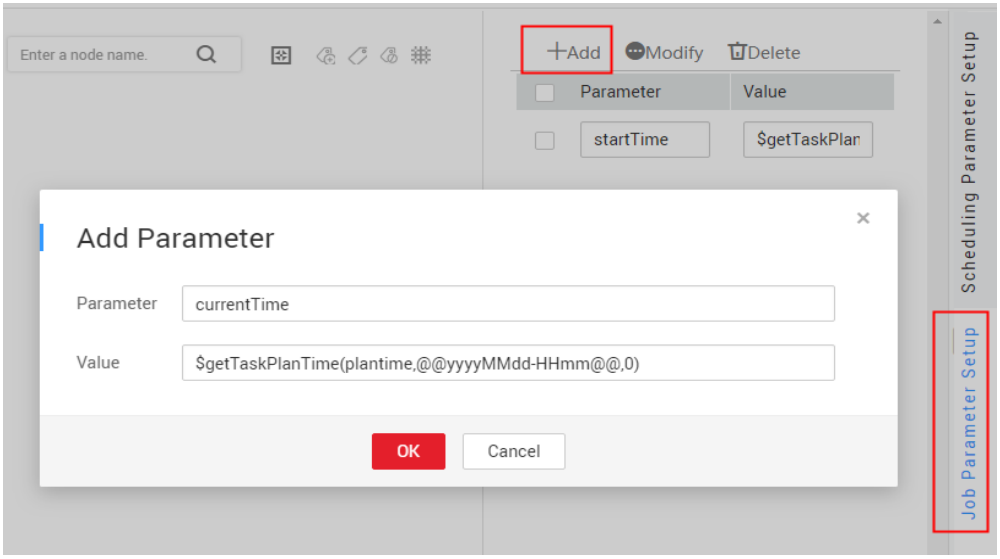
Figure 8-40 Properties of the node for deleting the CDM job



Configuring DLF Job Parameters

1. Configure the DLF job parameters. See **Figure 8-41**.
   - startTime = $getTaskPlanTime(plantime,@@yyyyMMddHHmmss@@,-24*60*60)
   - currentTime = $getTaskPlanTime(plantime,@@yyyyMMdd-HHmm@@,0)

Figure 8-41 Configuring DLF job parameters



2. After saving the DLF job, choose **Scheduling Configuration** > **Schedule periodically** and set the value to one day.

   In this way, DLF works with CDM to migrate data generated on the previous day.

8.5 Entire Database Migration to the Cloud

8.5.1 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 Log in to the **CDM management console** and create a CDM cluster. For details about how to create a CDM cluster, see **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

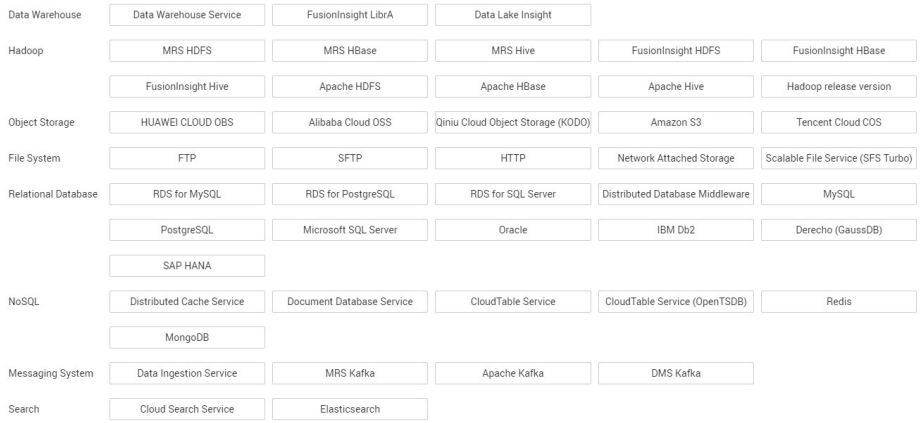
If SSL encryption is configured for the access channel of a local data source, CDM cannot connect to the data source using the EIP.

----End

Creating a Cloud Search Service Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the page that is displayed, choose **Link Management** > **Create Link**. The page for selecting a connector is displayed. See **Figure 8-42**.

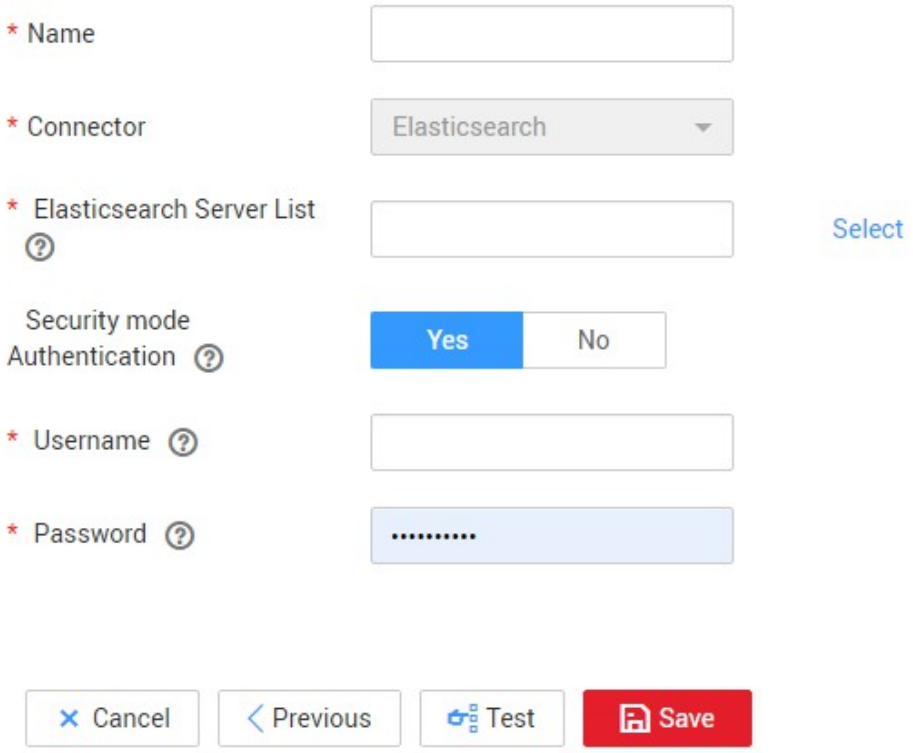
Figure 8-42 Selecting a connector



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 8-43 Creating a CSS link



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

----End

Creating an Elasticsearch Link

Step 1 On the **Link Management** tab page, click **Create Link**. On the page that is displayed, select **Elasticsearch**, click **Next**, and configure the Elasticsearch link parameters. The Elasticsearch link parameters are the same as those of the Cloud Search Service 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.
- **Username** and **Password**: If the Elasticsearch database has user restrictions, select the user who has the read and write permissions on the Elasticsearch database. If there is no user restriction, you do not need to configure the two parameters.

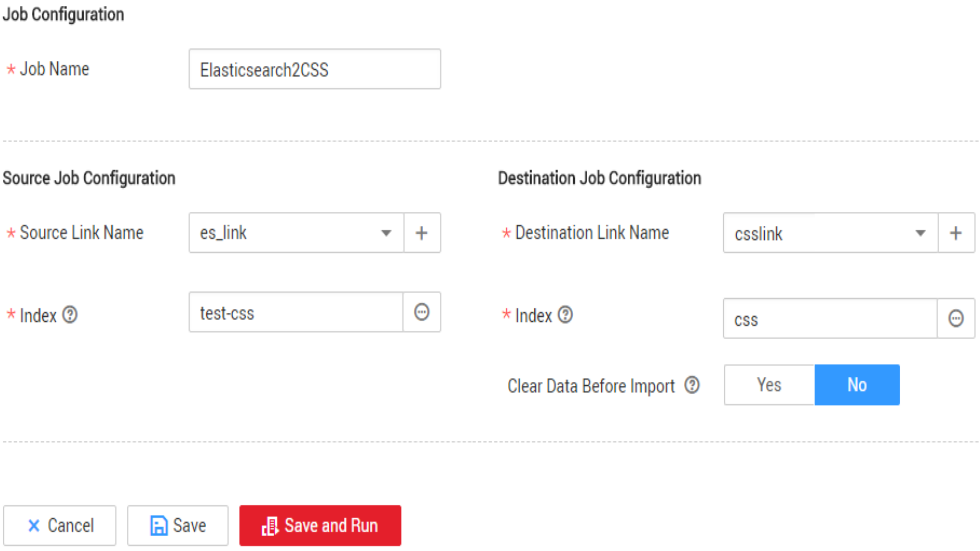
Step 2 Click **Save**. The **Link Management** page is displayed.

----End

Creating an Entire DB Migration Job

Step 1 Choose **Entire DB Migration** > **Create Job** to create an entire DB migration job.

Figure 8-44 Creating an entire DB migration job



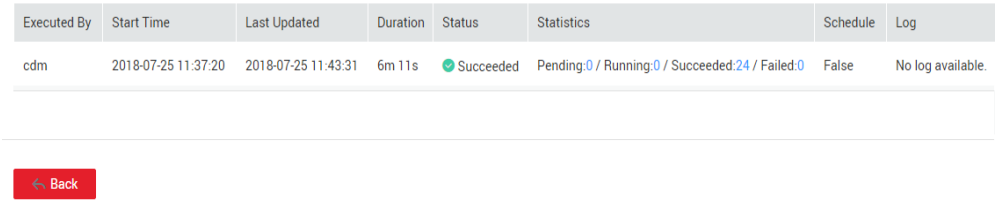
- **Job Name**: Enter a unique name.
- **Source Job Configuration**
  - **Source Link Name**: Select the **es_link** link created in **Creating an Elasticsearch Link**.
  - **Index**: Click the icon next to the text box to select an index in the on-premises Elasticsearch database or manually enter an 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 8-45 Historical Record



----End

9 Advanced Operations

9.1 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:

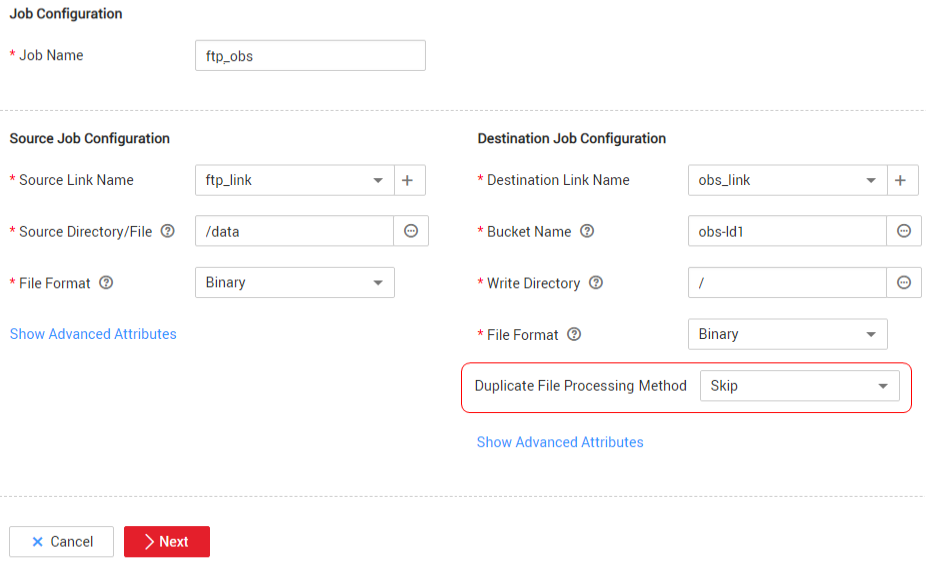
1. **Exporting all new files**
   - Application scenarios: Both the migration source and destination are file systems (OBS/OSS/HDFS/FTP/SFTP/NAS).
   - Key configurations: **Skipping Duplicate Files** and **Schedule Execution**
   - Prerequisites: None
2. **Exporting the files in a specified directory**
   - Application scenarios: The migration source is a file system (OBS/OSS/HDFS/FTP/SFTP/NAS). The migration destination can be of any type. 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.
3. **Exporting the files modified after the specified time point**
   - Application scenarios: The migration source is a file system (OBS/OSS/HDFS/FTP/SFTP/NAS). The migration destination can be of any type. The specified time point refers to the time when the file is modified. CDM migrates the files modified after the specified time point.
   - Key configurations: **Time Filter** and **Schedule Execution**
   - Prerequisites: None

Skipping Duplicate Files

- Parameter position: When **creating a table/file migration job**, if the migration source and destination are file systems, set **Duplicate File Processing Method** in **Destination Job Configuration** to **Replace**, **Skip**, or **Stop job**.

- Parameter principle: If a file with the same name and size exists on the migration source and destination, CDM determines that the file is a duplicate file.
- Example configurations:
  a. **Source Directory/File**: If you set this parameter to a directory, CDM imports all files in the directory to the migration destination.
  b. **File Format**: Select **Binary**. CDM directly copies the files without resolving the content, which is applicable to the migration of files to files.
  c. **Duplicate File Processing Method**: Select **Skip**.

  Figure 9-1 Skipping duplicate files



  d. Configure scheduled job execution.

In this way, you can import the newly added files to the destination directory periodically to implement incremental synchronization.

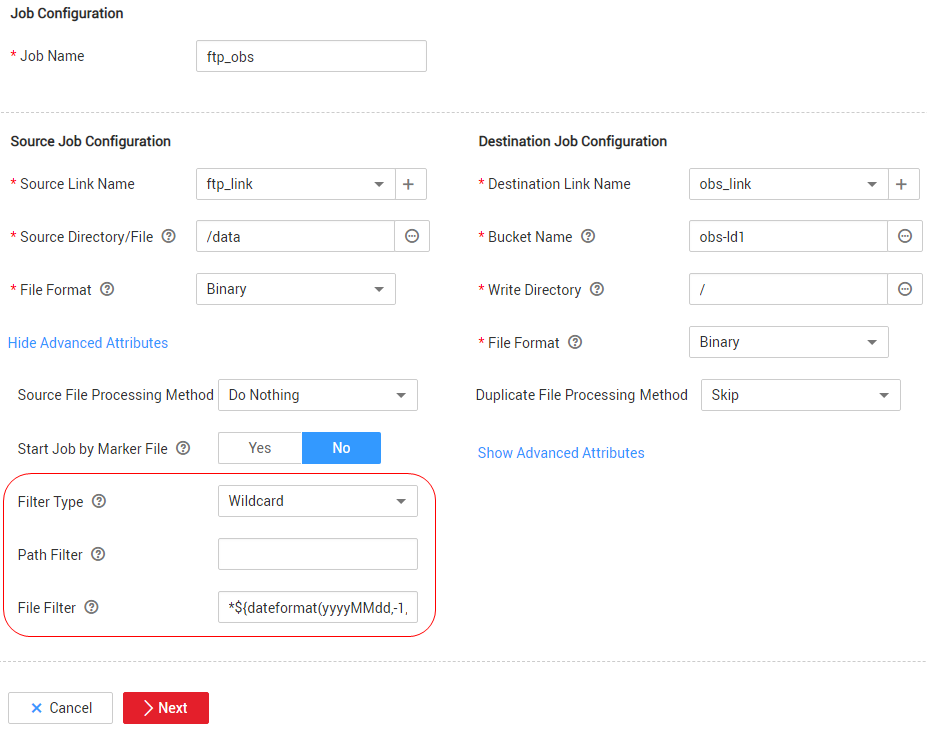
File/Path Filter

- 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 **Yes** for **Wildcard**, 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 **Incremental Synchronization Using the Macro Variables of Date and Time**.

Figure 9-2 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

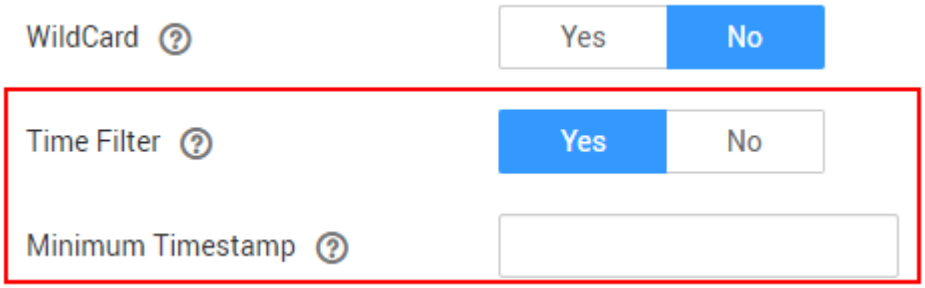
- Parameter position: When **creating a table/file migration job**, if the migration source is a file system, set select **Yes** for **Time Filter**.
- Parameter principle: If you specify **Modification Time**, only the files whose modification time is later than the specified time are migrated to CDM.
- Example configurations:

  Suppose that you want CDM to synchronize only the files generated after November 2, 2018 to the migration destination, configure the following parameters:

  a. **Time Filter**: select **Yes**.

  b. **Modification Time**: Enter a value in the format of *yyyy-MM-dd HH:mm:ss*, such as **2018-01-01 00:00:00**.

Figure 9-3 Time Filter



c. **Duplicate File Processing Method**: Select **Skip**.

d. Configure scheduled job execution.

In this way, the CDM job migrates only files generated after January 1, 2018, and performs incremental synchronization next time it is started.

9.2 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. For details, see **From 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.

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 9-4**. Set the parameters as follows:

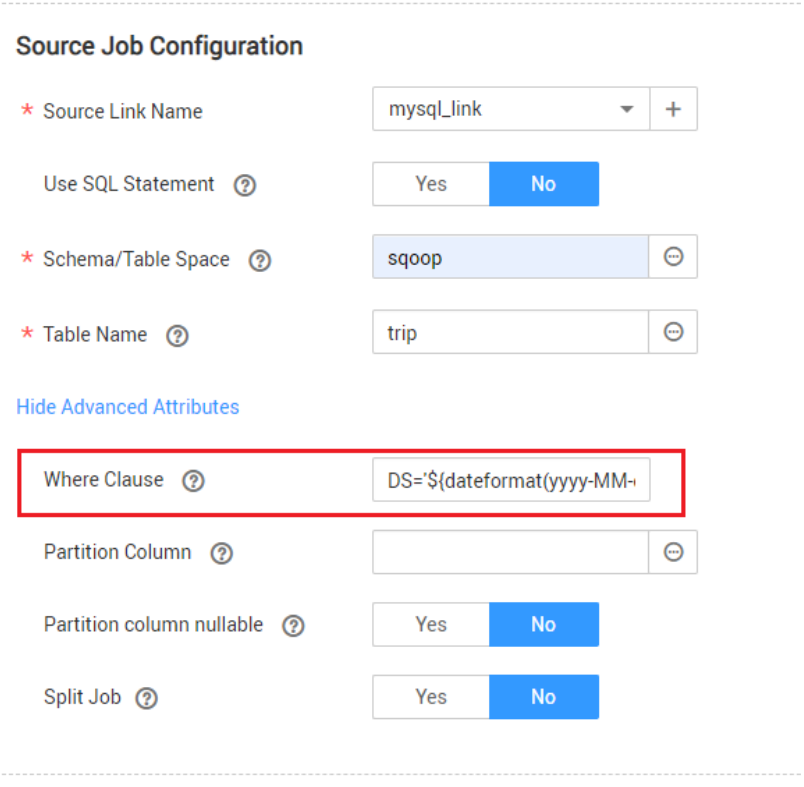
Figure 9-4 Table data

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 9-5 WHERE Clause



b. **Scheduling Job Execution**: Set **Cycle (days)** to **1** and **Start Time** to **00:00:00**.

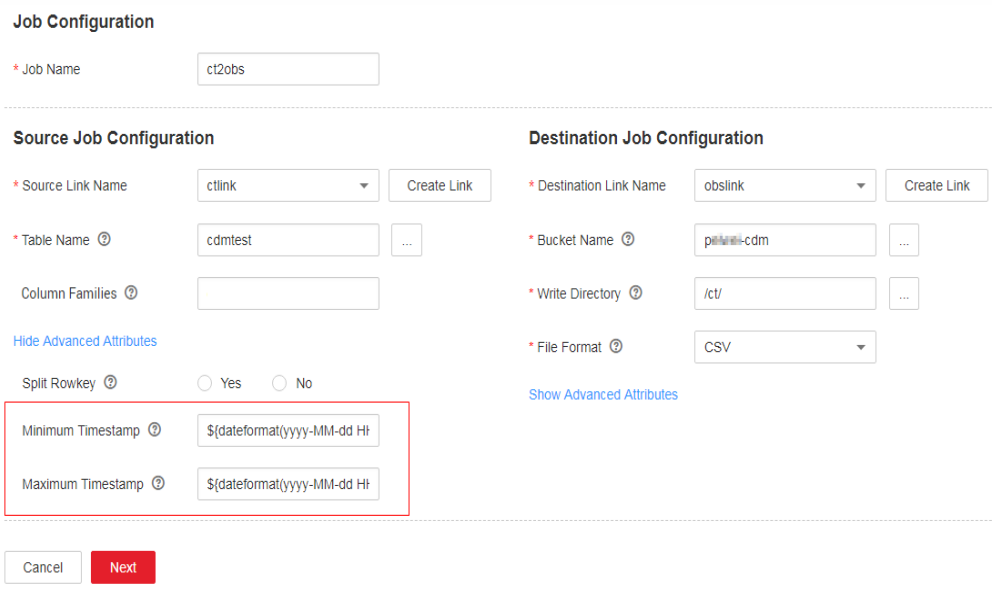
In this way, all data generated on the previous day can be exported at 00:00:00 every day. **WHERE Clause** can be configured to various **macro variables of date and time**. You can use the macro variables of date and time and scheduled jobs with specified cycle of minutes, hours, days, weeks, or months together to automatically export data at a specific time.

9.3 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.

When creating a table/file migration job and selecting **Link to HBase** or **Link to CloudTable** as the source link, you can set the time range in advanced attributes.

Figure 9-6 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.

9.4 Incremental Synchronization Using the 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
- Source table name
- Write directory
- Destination table name
- Where clause

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.

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, and DAY.

  For example, if the current date is **2017-10-16 09:00:00**, **dateformat(yyyy-MM-dd HH:mm:ss, -1, DAY)** indicates the day before the current day, that is, **2017-10-15 09:00:00**.

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 9-1** describes the macro variable definitions of time and date.

Table 9-1 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

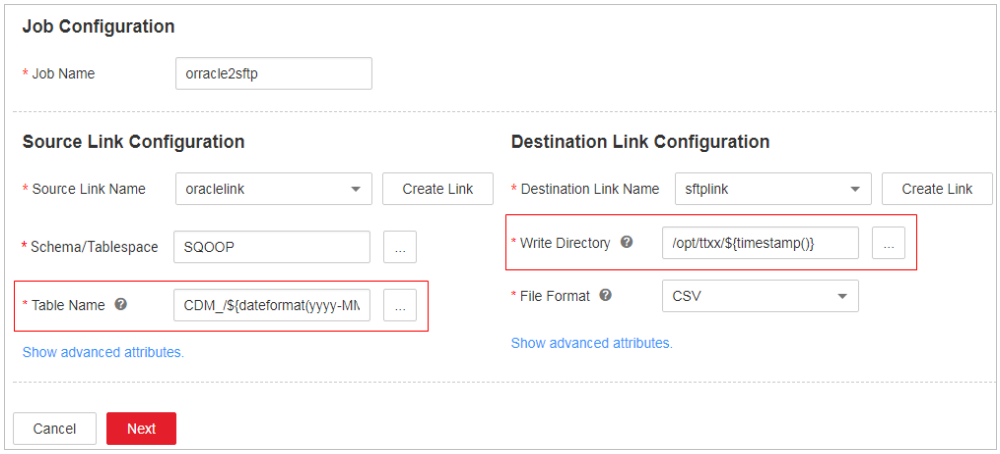
Time and Date Macro Variables of Paths and Table Names

Figure 9-7 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 SFTP server.

Figure 9-7 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 9-8 uses table **SQOOP.CDM_20171016** as an example. The table contains column **DS**, which indicates the time.

Figure 9-8 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 in the previous day will be exported at 00:00:00 every day.

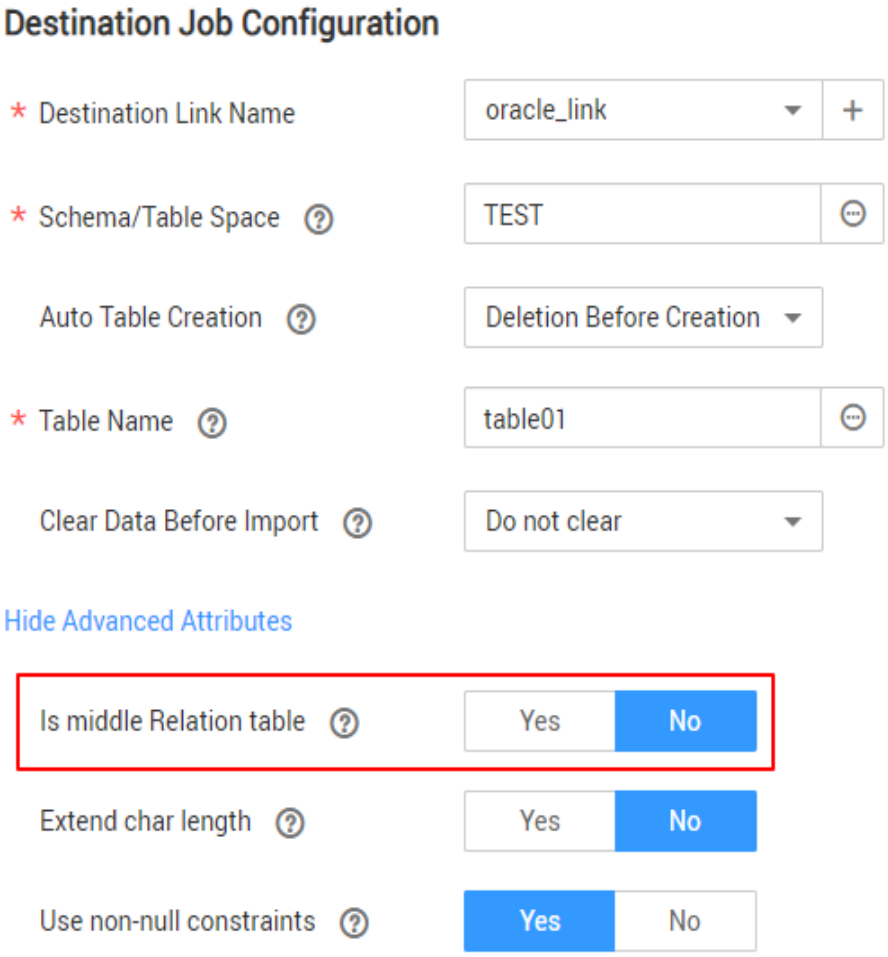
Configuration principles of other application scenarios are the same.

9.5 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 9-9 Migration in transaction mode



NOTE

If you set **Clear Data Before Import** to **Yes**, CDM does not roll back the deleted data even in transaction mode.

9.6 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: OBS, FTP, SFTP, NAS, SFS, HDFS (supported in the binary format), and HTTP (applicable to scenarios where OBS shared files are downloaded)
- Data sources supported by the migration destination: OBS, FTP, SFTP, NAS, SFS, and HDFS (supported in the binary format)

The following describes how to use the AES-256-GCM algorithm for encryption and decryption in OBS file migration. The methods for using the algorithm on other data sources are the same.

- **Configure decryption at the migration source.**

  When you use CDM to create a job for exporting files from OBS, set the migration source to OBS 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 OBS, the files written to the migration destination are decrypted plaintext files.

- **Configure encryption at the migration destination.**

  When you use CDM to create a job for importing files to OBS, set the migration destination to OBS and set the following parameters in the advanced settings of **Destination Job Configuration**:

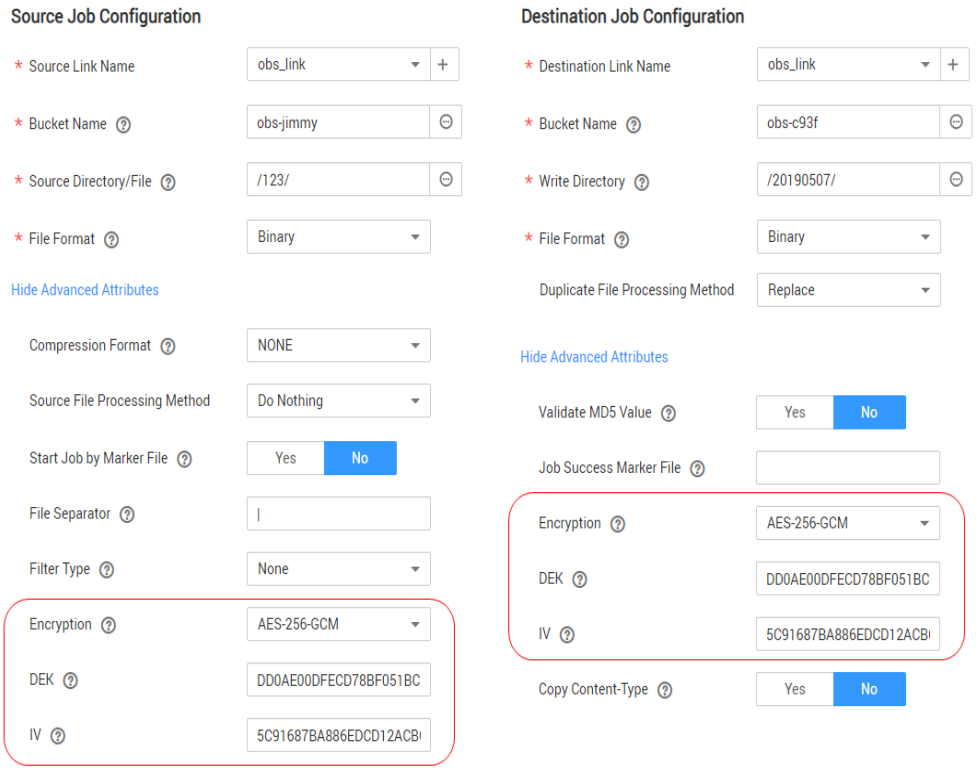
  a. **Encryption**: Select **AES-256-GCM**.

  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 OBS, the files on the migration destination are encrypted using the AES-256-GCM algorithm.

Figure 9-10 AES-256-GCM encryption and decryption



KMS Encryption

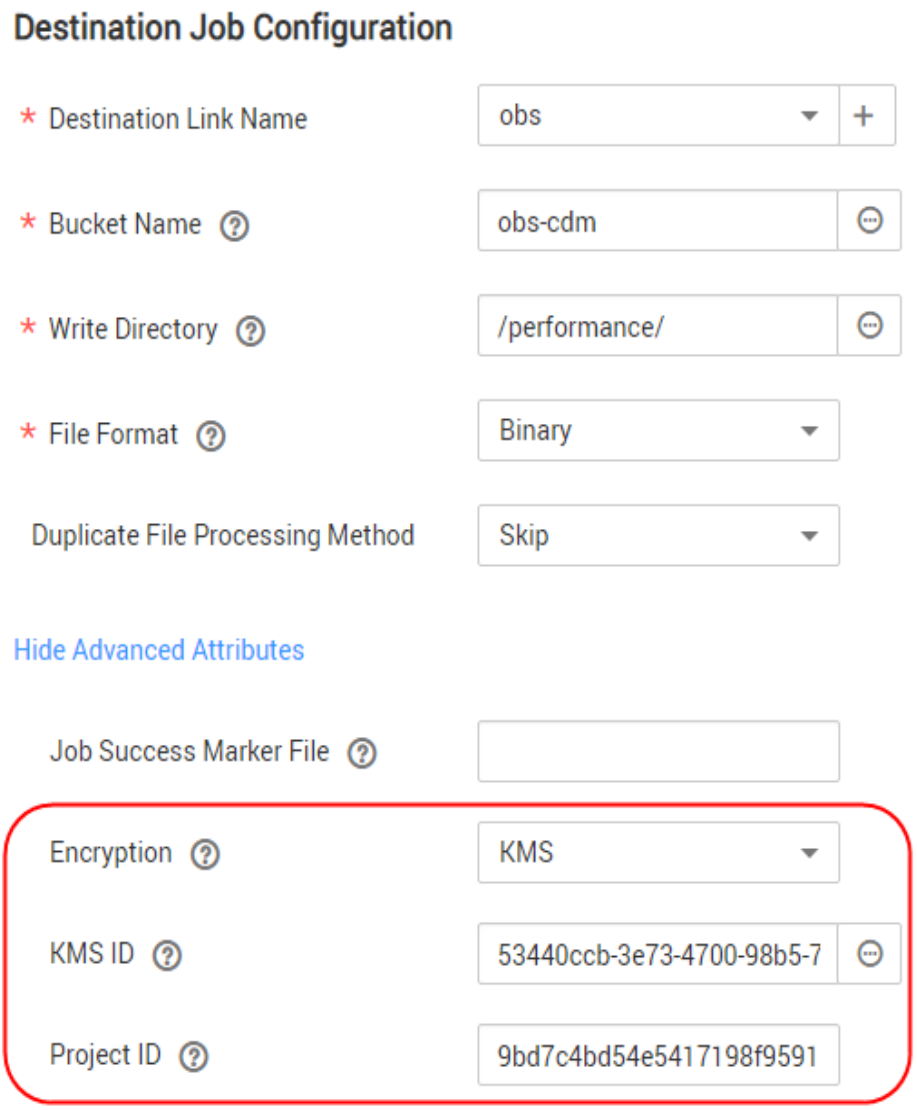
NOTE

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. See **Figure 9-11**.

A key must be created in KMS of DEW in advance. For details, see the *Data Encryption Workshop User Guide*.

Figure 9-11 Enabling KMS encryption



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.

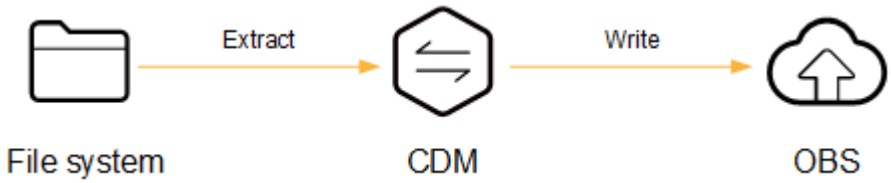
NOTE

- 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.

9.7 MD5 Verification

CDM extracts data from the migration source and writes the data to the migration destination. **Figure 9-12** shows the migration mode when files are migrated to OBS.

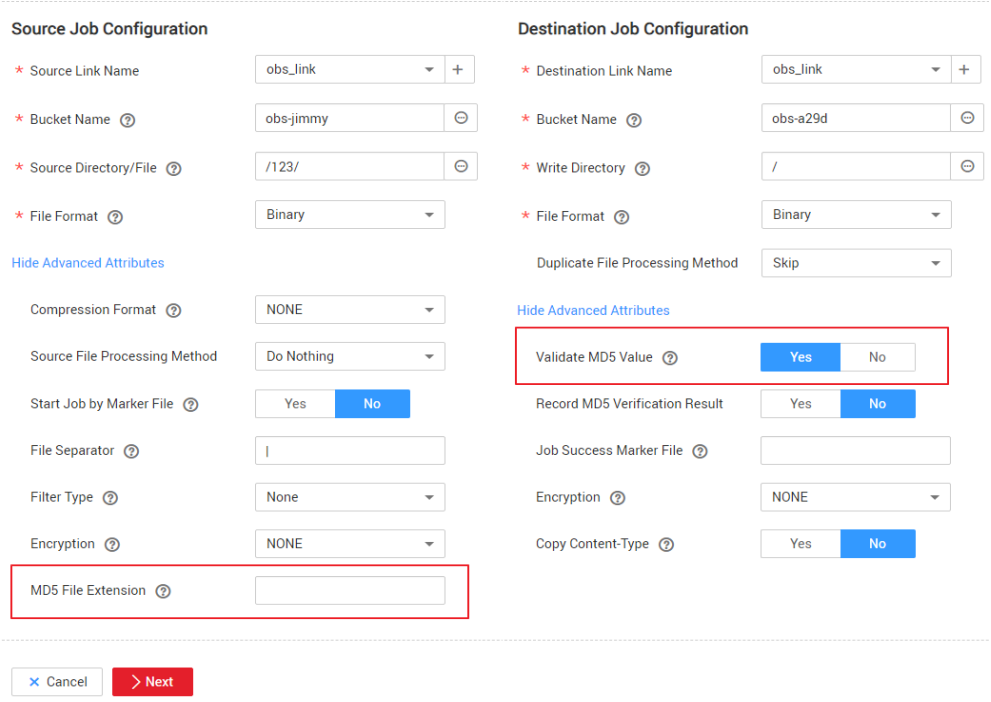
Figure 9-12 Migrating files to OBS



During the process, CDM uses MD5 to verify file consistency. For details about the parameters, see **Figure 9-13**.

- **Extract**
  - Check whether the files extracted by CDM are consistent with source files.
  - This function is controlled by the **MD5 File Extension** parameter 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.
  - The migration source can be OBS, FTP, SFTP, NAS, SFS, or HTTP.
- **Write**
  - 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.
  - Currently, this function can be used only when OBS serves as the migration destination.

Figure 9-13 Enabling MD5 verification to verify file consistency



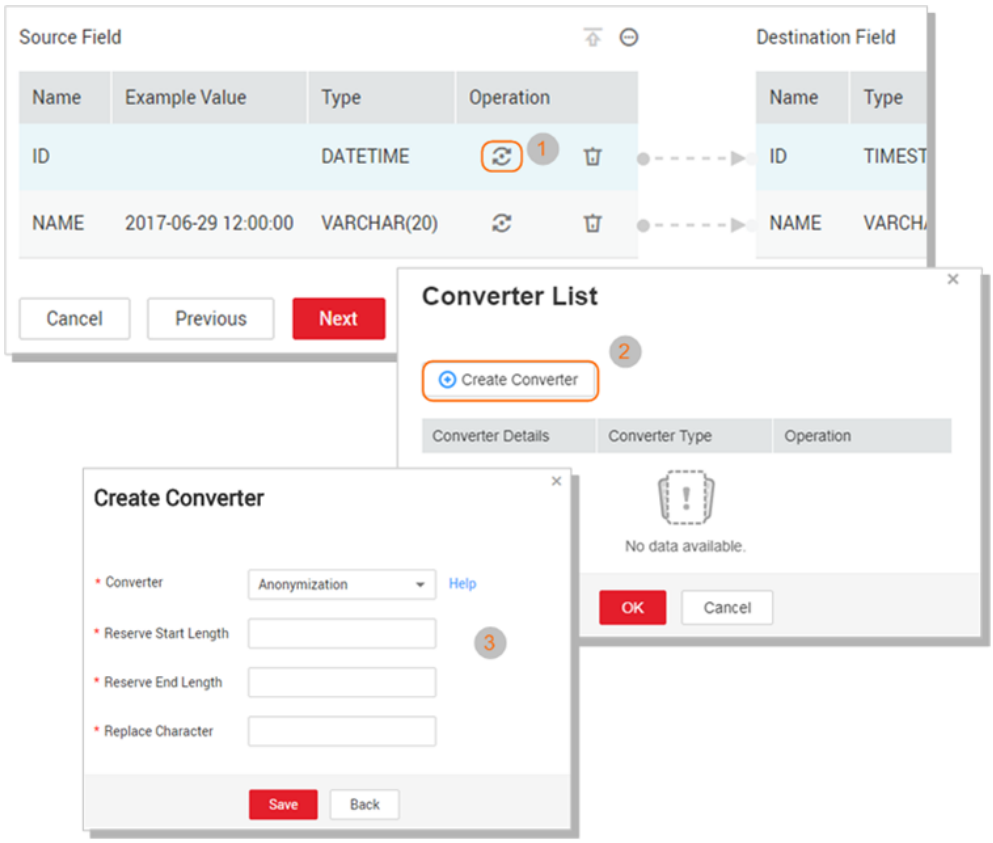
NOTE

- 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.

9.8 Field Conversion

You can create a field converter on the **Map Field** tab page when creating a table/file migration job. See **Figure 9-14**.

Figure 9-14 Creating a field converter



NOTE

Field mapping is not involved when the binary format is used to migrate files to files.

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

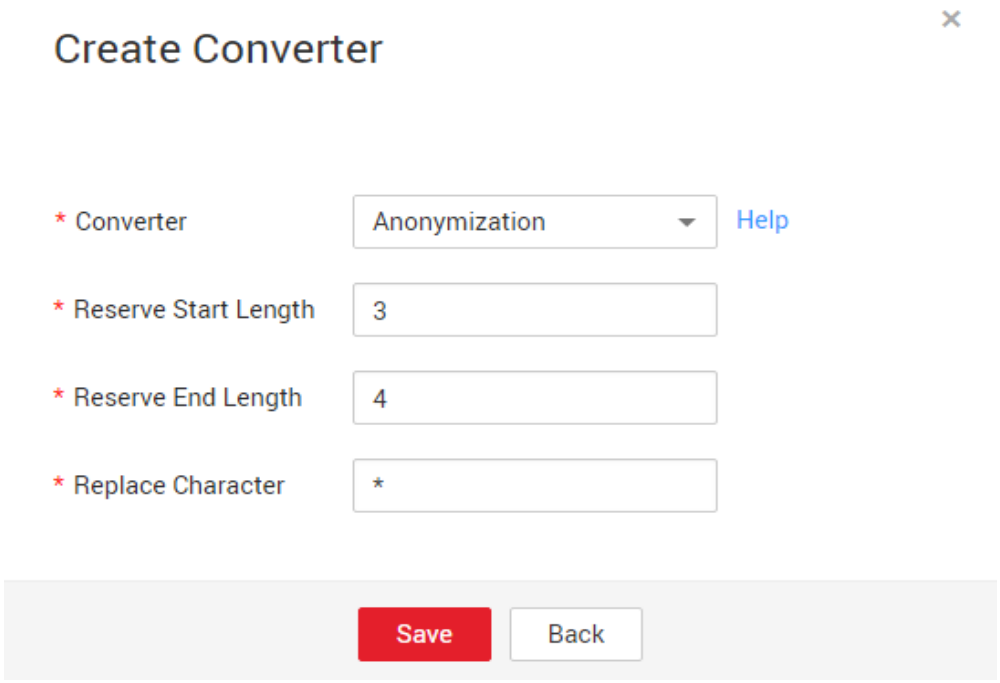
- **Anonymization**
- **Trim**
- **Reverse String**
- **Replace String**
- **Remove line break**
- **Expression Conversion**

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 *****.

Figure 9-15 Anonymization



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 tool classes:

- StringUtils: string processing tool class. For details, see **org.apache.commons.lang.StringUtils** of the Java SDK code.
- DateUtils: date tool class
- CommonUtils: common tool class
- NumberUtils: string-to-value conversion class
- HttpsUtils: network file read class

Application examples:

1. Set a string constant for the current field, for example, VIP.
   Expression: **"***VIP***"**
2. 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)
3. Convert all character strings of the current field to uppercase letters.
   Expression: StringUtils.upperCase(value)
4. 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,"-")
5. 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
6. Convert the field value **true** to **Y** and other field values to **N**.
   Expression: value=="true"?"Y":"N"
7. 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
8. If the first and second fields are of the numeric type, convert the field to the combination of the first and second field values.
   Expression: row[0]+row[1]
9. If the field is of the date or timestamp type, return the current year after conversion. The data type is int.
   Expression: DateUtils.getYear(value)
10. If the field is a date and time string in *yyyy-MM-dd* format, convert it to the date type:
    Expression: DateUtils.format(value,"yyyy-MM-dd")
11. 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")
12. Obtain a 36-bit universally unique identifier (UUID):
    Expression: CommonUtils.randomUUID()
13. If the field is of the string type, capitalize the first letter, for example, convert **cat** to **Cat**.

Expression: StringUtils.capitalize(value)

14. 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)

15. 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*)

16. 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)

17. 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*")

18. 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*")

19. 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*")

20. 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*")

21. 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*)

22. 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*)

23. 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*")

24. 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*")

25. 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*")

26. 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*)

27. 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*")

28. 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)

29. 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)

30. 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)

31. 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)

32. 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)

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

    Expression: StringUtils.isEmpty(value)

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

    Expression: StringUtils.isNumeric(value)

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

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

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

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

37. 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*")

38. 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*")

39. 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)

40. 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*")

41. 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*")

42. 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.*")

43. 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*")

44. 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*")

45. If the field is of the string type, use the specified delimiter to split the text into arrays. For example, use **:** to split **ab:cd:ef** into **["ab","cd","ef"]**.

    Expression: StringUtils.split(value,":")

46. 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*")

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

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

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

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

49. 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*)

50. If the field is of the string type, obtain the substring after the specified position (excluding 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. For example, obtain the character string after the second character of **abcde**, that is, **cde**.

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

51. If the field is of the string type, obtain the substring within the specified range of the character string. If the specified range is a negative number, calculate the range in the descending order. For example, obtain the character string between the second and fifth characters of **abcde**, that is, **cd**.

    Expression: StringUtils.substring(value,*2*,*5*)

52. 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*")

53. 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*")

54. 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*")

55. 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*")

56. 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*")

57. 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)

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

    Expression: NumberUtils.toByte(value)

59. 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*)

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

    Expression: NumberUtils.toDouble(value)

61. 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*)

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

    Expression: NumberUtils.toFloat(value)

63. 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*)

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

    Expression: NumberUtils.toInt(value)

65. 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*)

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

    Expression: NumberUtils.toLong(value)

67. 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*)

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

    Expression: NumberUtils.toShort(value)

69. 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*)

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

    Expression: CommonUtils.ipToLong(value)

71. 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*")

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

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

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

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

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

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

75. Obtain the physical addresses corresponding to the IP address in *Country_Province_City_Carrier* format. For example, the physical address corresponding to **1xx.78.124.0** is **China_Guangdong_Shenzhen_China Telecom**. If the corresponding physical address cannot be obtained, the

default value **_**_**_** is returned. If necessary, you can use the StringUtil class expression to further split the addresses.

Expression: CommonUtils.getMapValue(CommonUtils.ipToLong(value),CommonUtils.cacheExists("*ipLis*")?CommonUtils.getCache("*ipLis*"):CommonUtils.setCache("*ipLis*",HttpsUtils.downloadMap("*url*")))

76. 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**.

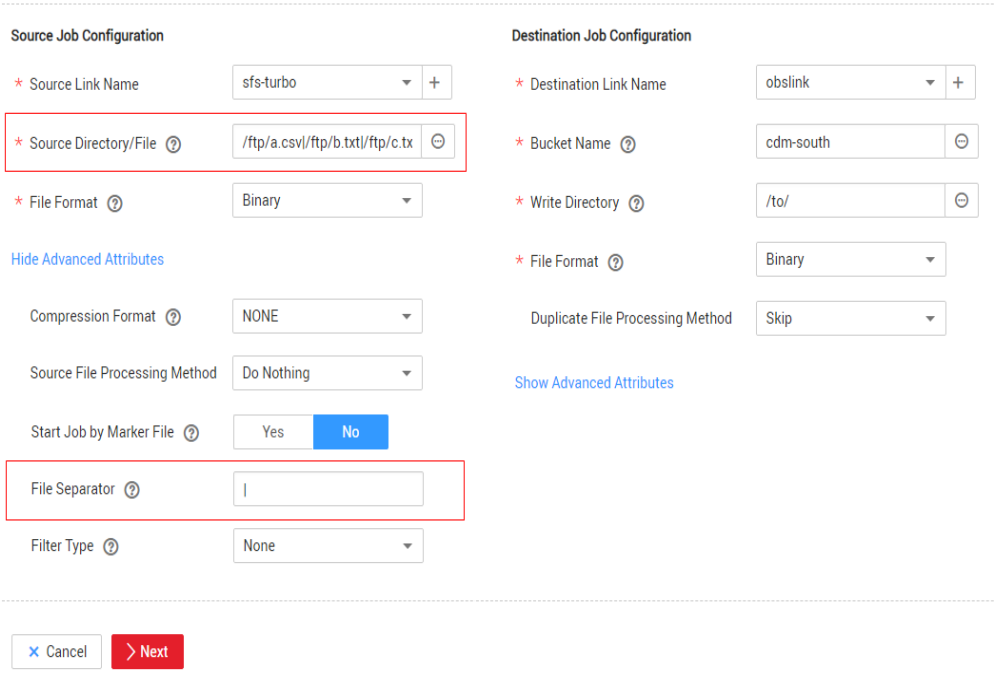
    Expression: DateUtils.getCurrentTimeByZone("*yyyy-MM-dd HH:mm:ss*",value,"*hour*", *8*)

9.9 Migration of a List of Files

You can migrate a list of files (a maximum of 50 files) from FTP, SFTP, NAS, OBS, OSS, KODO, or SFS 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, SFTP, NAS, OBS, OSS, KODO, or SFS, **Source Directory/File** can contain a maximum of 50 file names, which are separated by vertical bars (|). You can also customize a file separator.

Figure 9-16 Migrating a list of files



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.

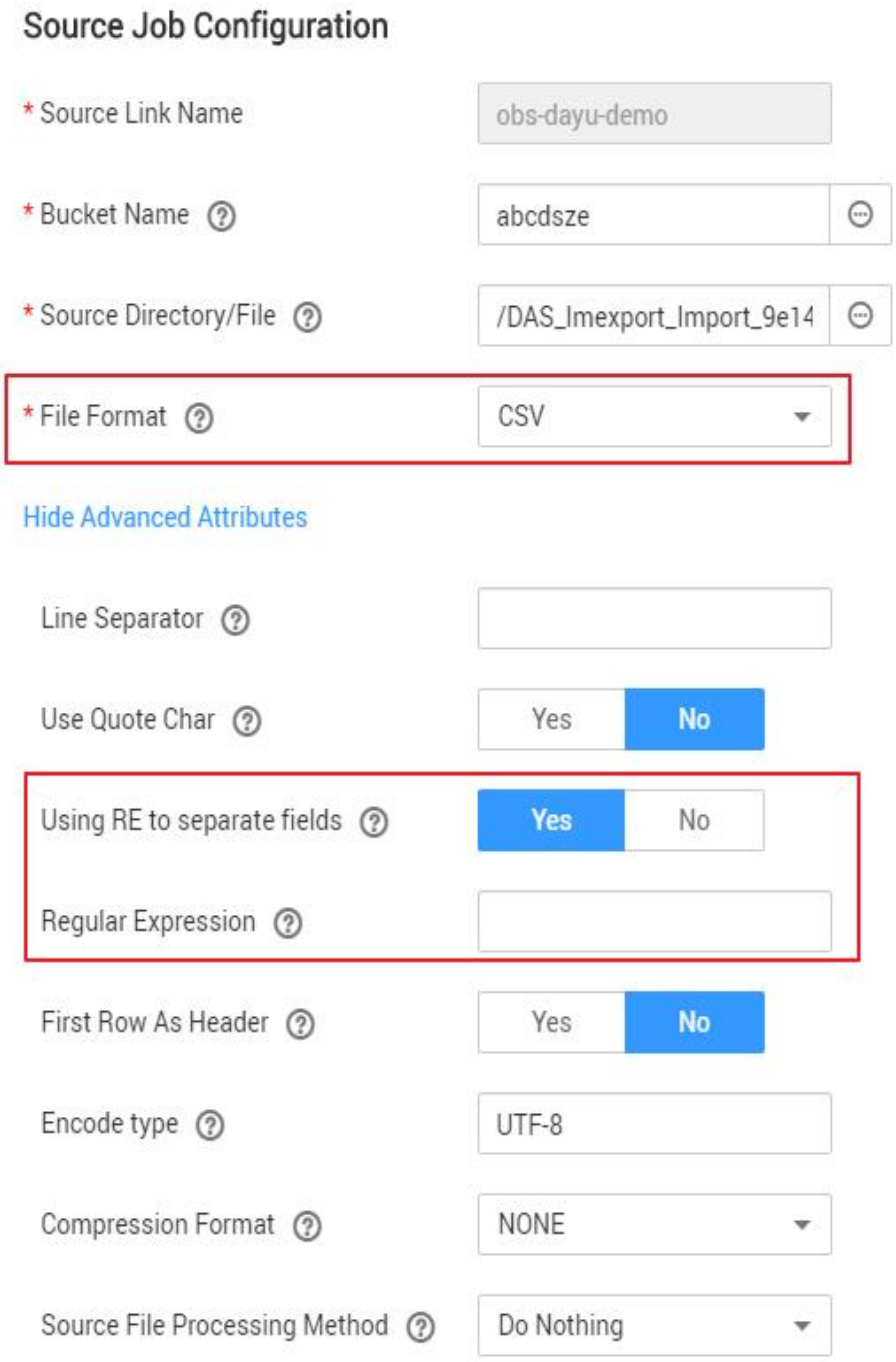
2. During file migration, a single task supports millions of files. If there are too many files in the directory to be migrated, you are advised to split the files into different directories and create multiple tasks.

9.10 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**.

Figure 9-17 Setting regular expression parameters



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 9-2 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 9-3 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

Column Number	Example Value
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 9-4 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 9-5 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 9-6 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

9.11 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 9-7 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 9-7**.

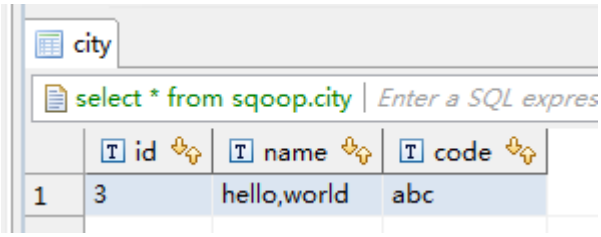
3. **Encoding Type**

   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 9-18** shows that the value of the **name** field in the database contains a comma (,).

     Figure 9-18 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": ["chinese","english","math"]
        }
      },
      {
          "_id": "650616",
          "_source": {
              "name": "tom",
              "books": ["chinese","english","math"]
          }
      },
     {
          "_id": "650618",
          "_source": {
              "name": "tom",
              "books": ["chinese","english","math"]
```



```
        }
      }]
   }
}
```

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	["chinese","english","math"]
650616	tom	["chinese","english","math"]
650618	tom	["chinese","english","math"]

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

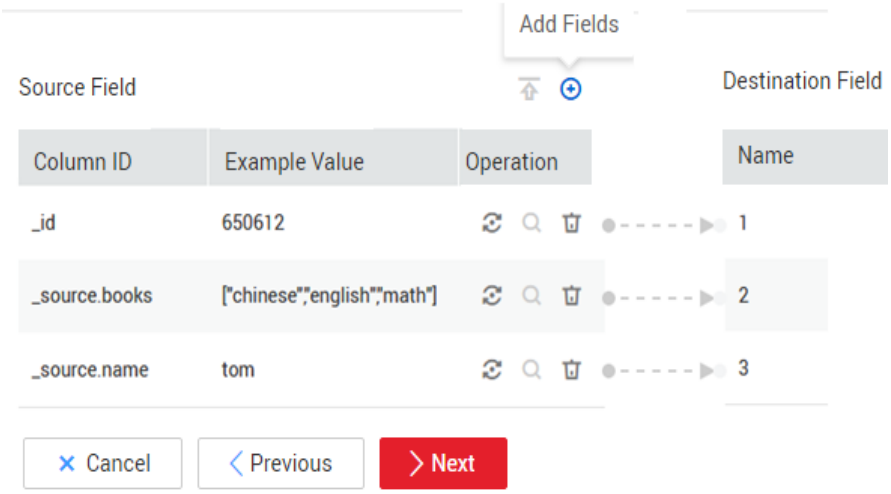
d. Example 4: Configure a converter when parsing the JSON file. On the premise of **example 2**, to add the **hits.max_score** field to all records, that is, to write the data to the database in the following formats, perform the following operations:

ID	SourceName	SourceBooks	MaxScore
650612	tom	["chinese","english","math"]	1.0
650616	tom	["chinese","english","math"]	1.0
650618	tom	["chinese","english","math"]	1.0

Set **File Format** to **JSON**, **JSON Type** to **JSON object**, and **JSON Reference Node** to **hits.hits**, and then create a converter.

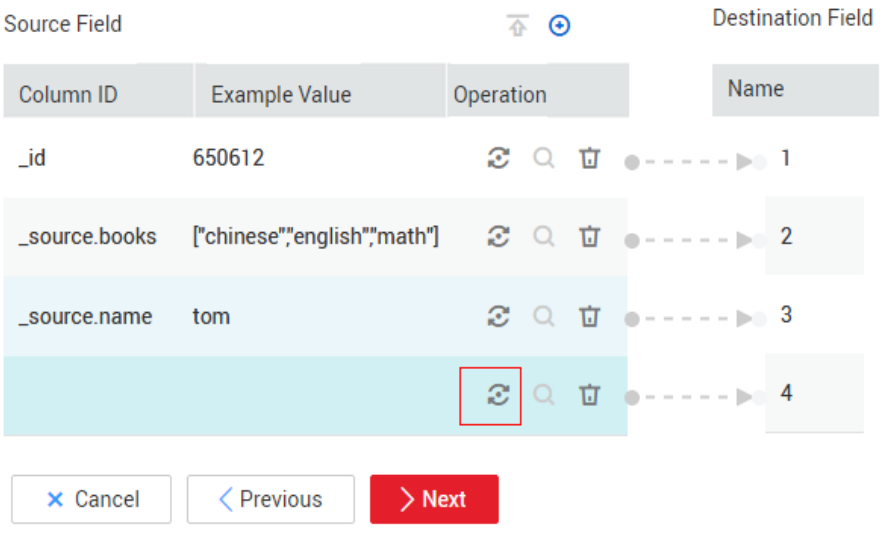
i. Click ⊕ to add a field.

Figure 9-19 Adding a field



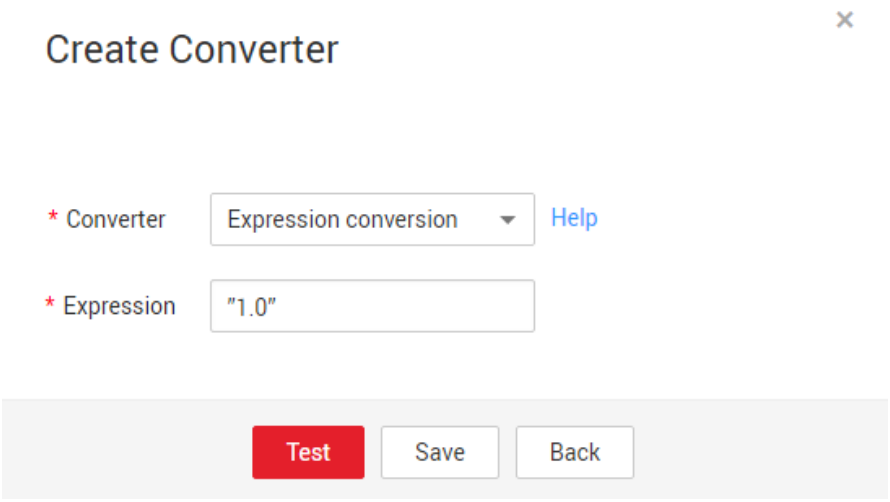
ii. Click ⟳ to create a converter for the new field.

Figure 9-20 Creating a field converter



iii. Set **Converter** to **Expression conversion**, enter **"1.0"** in the **Expression** text box, and click **Save**.

Figure 9-21 Configuring a field converter



Binary

If you want to copy files between file systems, you can select the binary format. The binary format delivers the optimal rate and performance in file transfer, and does not require field mapping.

- **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

- **Source File Processing Method**

  After a file is copied successfully, CDM can perform operations on the source file. The options are **Rename** and **Delete**.

- **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 9-7**.

   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.

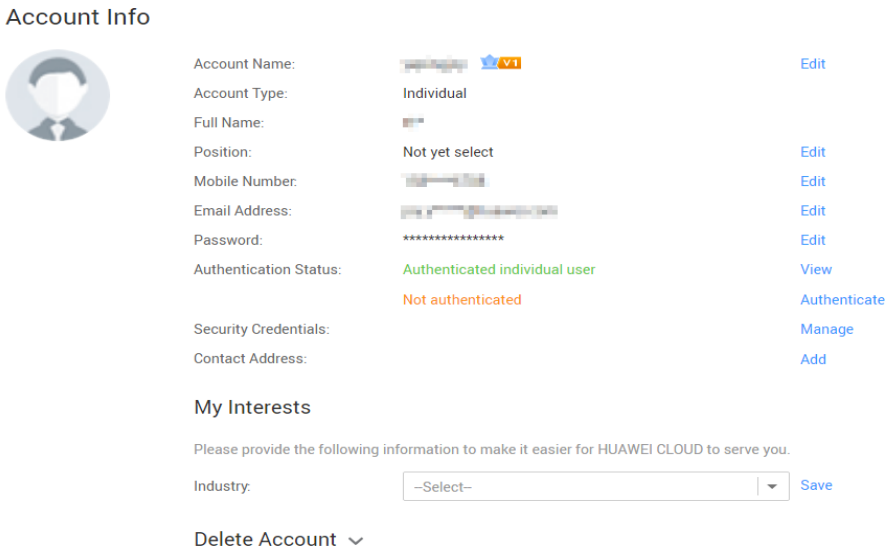
10 Appendix

10.1 Obtaining Authentication Information

Obtaining an AK/SK Pair

1. Log in to the management console, move your pointer to the username in the upper right corner of the page, and select **Basic Information** from the drop-down list.
2. On the **My Account** page, choose **Account Info** > **Manage**. See **Figure 10-1**.

   Figure 10-1 Account Info page

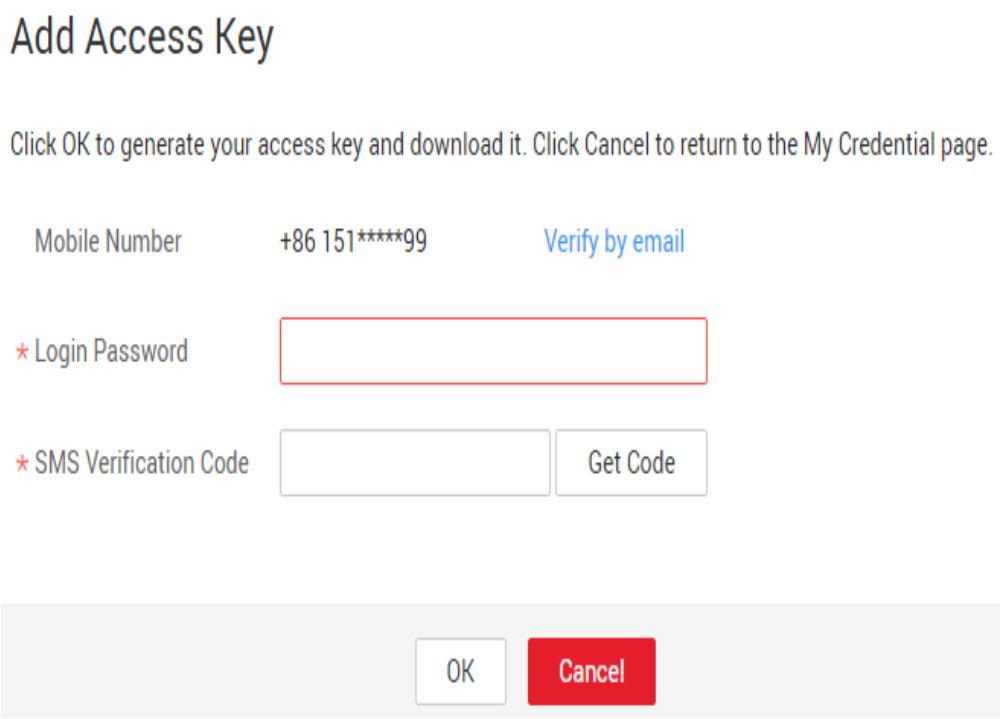
   

3. On the **My Credentials** page, choose **Access Keys** > **Add Access Key**. See **Figure 10-2**.

Figure 10-2 Access Keys page

4. In the **Add Access Key** dialog box, select where the verification code will be sent, and enter the verification code. See **Figure 10-3**.

Figure 10-3 Add Access Key dialog box



5. Click **OK** and save the access key file. The access key file is saved to your browser's configured download location. Open the **credentials.csv** file to view the AK and SK.

Obtaining a Project ID

A project is a group of user resources. To view the project IDs of different regions, choose **My Credentials** > **Projects**.

Obtaining a Region and Endpoint

For details on regions and endpoints, see **Regions and Endpoints**.

A Change History

Release Date	Description
2020-11-20	This is the twenty-ninth official release. Deleted the following migration cases: • From Redis to DCS • From OSS to SFS • Migrating the Entire MySQL Database to RDS
2020-10-23	This is the twenty-eighth official release. Added **Managing Drivers**.
2020-06-18	This is the twenty-seventh official release. Modified **To a Relational Database**.
2020-05-11	This is the twenty-sixth official release. Add metrics to section **CDM Metrics**. Added **Scenario-based Migration**. Added the cdm.4xlarge flavor to section **Creating a CDM Cluster**. Supported entire database migration from Hive to DWS. For details, see **Entire DB Migration**.
2020-02-06	This is the twenty-fifth official release. Modified the method of viewing metrics in section **Configuring Alarm Rules**.
2020-01-20	This is the twenty-fourth official release. Modified **Permissions Management**.
2020-01-03	This is the twenty-third official release. Modified **Permissions Management**.

Release Date	Description
2019-11-25	This is the twenty-second official release. • Added **Agent Management** because CDM provides agent management. • Modified **Creating a User and Granting CDM Permissions** and **Creating a Custom Policy**.
2019-11-04	This is the twenty-first official release. • Modified **From OBS/OSS/KODO/COS/S3**, **From HDFS**, and **From FTP/SFTP/NAS/SFS** because CDM allows you to filter files by wildcard and file modification time. • Modified **To OBS** because the storage directory of migrated files can be customized. • Modified **From HDFS** because files can be migrated in batches from HDFS.
2019-09-22	This is the twentieth official release. Modified **MD5 Verification**.
2019-07-02	This is the nineteenth official release. • Updated **Creating Links**, **Link to KODO/COS**, and **Table/File Migration Jobs** because data can be exported from COS to OBS. • Updated **Job Configuration Management** because jobs can be automatically backed up to OBS buckets. • Updated **From OBS/OSS/KODO/COS/S3**, **From HDFS**, and **From FTP/SFTP/NAS/SFS** because historical files can be migrated based on the creation time.

Release Date	Description
2019-06-06	This is the eighteenth official release. • Added **IAM Permissions Management**. • Modified **Entire DB Migration** because the entire DB migration of CSS/Elasticsearch supports multi-index migration. • Modified **Entire DB Migration** because concurrent migration of Oracle partition tables is supported. • Modified **From a Relational Database** because when exporting data from a MySQL database, you can migrate all tables starting with a certain prefix (the number and types of fields in the tables must be the same) to a Hive table. • Modified **To OBS** because when data is imported to OBS, CDM allows you to use KMS keys of other projects to encrypt data. • Modified **To OBS** because when importing data to OBS, you can customize the file names. • Added **Job Configuration Management** because global variables are supported. • Modified **MD5 Verification** because CDM supports file consistency check. • CDM can connect to the MRS 2.0 cluster.
2019-04-03	This is the seventeenth official release. • Modified **Managing Jobs in Batches** because job grouping is supported. • Modified **Modifying Cluster Configurations** because job sharing among multiple IAM users under an account is supported. • Modified **Entire DB Migration** because the entire DB migration from MongoDB to DWS is supported. • Modified the following sections because the AES-256-GCM (NoPadding) algorithm can be used to encrypt and decrypt files in file migration: – **From OBS/OSS/KODO/COS/S3** – **From HDFS** – **From FTP/SFTP/NAS/SFS** – **From HTTP/HTTPS** – **To OBS** – **To HDFS** – **To FTP/SFTP/NAS/SFS** – **Encryption and Decryption During File Migration**

Release Date	Description
2019-01-30	This issue is the sixteenth official release. • Modified **From DIS** and **From Apache Kafka/DMS Kafka** because the binary format is supported in data migration of Kafka, DIS, and DMS. • Modified **From HTTP/HTTPS** because you can choose whether to delete the **query** parameter from the name of the objects uploaded to OBS. • Modified **Field Conversion** because the **Remove line break** converter is added for field conversion. • Modified **Entire DB Migration** because the migration source of the entire DB migration supports the WHERE clause
2019-01-07	This is the fifteenth official release. • Modified **Creating a CDM Cluster** because CDM allows you to configure multiple mobile numbers and email addresses. • Modified **Managing a Single Job** because the **Historical Jobs** tab page is added, allowing you to view historical configuration information about jobs and one-time jobs that are periodically executed, as well as re-execute the historical jobs. • Modified the parameters in **From HDFS** because when CDM extracts files from HDFS, a snapshot can be created for the HDFS source directory. • Modified the parameters in **To Elasticsearch or CSS** because 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.

Release Date	Description
2018-11-30	This is the fourteenth official release. • Added the following sections because CDM supports data migration of DMS Kafka and Amazon S3: – **Link to Amazon S3** – **Link to DMS Kafka** • Modified the following sections because the file separator can be customized and the heading line of the table can be written to the target file: – **From OBS/OSS/KODO/COS/S3** – **From FTP/SFTP/NAS/SFS** – **To OBS** – **To FTP/SFTP/NAS/SFS** • Modified **To a Relational Database** because during relational database migration, data in the destination table can be deleted based on the WHERE clause. • Modified **Entire DB Migration** because dirty data write is supported and the field length is automatically extended when data is written to DWS.
2018-10-30	This is the thirteenth official release. • Modified **To OBS** because after data is successfully imported to OBS, a job success marker file is written. • Added **To DIS**. • Modified **Modifying Cluster Configurations** because the CDM clusters can be started, stopped, restarted, or deleted in batches. • Added the following sections: – **Incremental Migration on CDM Supported by DLF**

Release Date	Description
2018-09-30	This is the twelfth official release. • Modified the following sections because CDM supports data migration of SFS: – **Creating Links** – **Link to NAS/SFS** – **Table/File Migration Jobs** – **From FTP/SFTP/NAS/SFS** – **To FTP/SFTP/NAS/SFS** • Added the bucket-level domain name to **Link to OBS**. • Added the function of using an SQL statement to export data to **From a Relational Database**. • Added the following sections: – **Link to CloudTable OpenTSDB** – **From OpenTSDB** – **To OpenTSDB**
2018-08-31	This is the eleventh official release. • Added SMS message and email notifications upon table/file migration job failures to **Creating a CDM Cluster** • Updated **Unbind and Release EIP** in **Binding or Unbinding an EIP**. • Added **To DIS**. • Modified the following sections because you can migrate multiple indexes at a time in entire DB migration of Elasticsearch or CSS: – **Link to Elasticsearch/CSS** – **Entire DB Migration** – **Migrating the Entire Elasticsearch Database to CSS** • Modified the following sections because the time filter function is added: – **From OBS/OSS/KODO/COS/S3** – **From HDFS** – **From FTP/SFTP/NAS/SFS** – **Incremental File Migration**

Release Date	Description
2018-08-03	This is the tenth official release. • Added the following sections: – **From OSS to OBS** – **From OBS to CSS** – **Migrating the Entire Elasticsearch Database to CSS** – **File Formats** • Updated the screenshots. • Modified the operation procedures in section "Typical Scenarios." • Modified the description of most job parameters in section "Job Management" and added multiple job parameters.
2018-07-05	This is the ninth official release. • Added the following sections: – **Link to KODO/COS** • Updated the screenshots. • Modified the parameter description in the following sections: – **Creating Links** – **Link to HDFS** – **Link to HBase** – **From Elasticsearch or CSS** – **To OBS** – **To FTP/SFTP/NAS/SFS**

Release Date	Description
2018-06-02	This is the eighth official release. • Added the following sections: – **From HTTP/HTTPS** – **From MySQL to MRS Hive** – **HBase/CloudTable Incremental Migration** • Updated the screenshots. • Modified the following sections because HTTP/HTTPS can be used as the migration source: – **Creating Links** – **Table/File Migration Jobs** • Modified the following sections because the auto shutdown and scheduled startup/shutdown are supported: – **Step 1: Creating a Cluster** – **Creating a CDM Cluster** – **Modifying Cluster Configurations** • Modified the parameter description in the following sections: – **Link to Elasticsearch/CSS** – **From OBS/OSS/KODO/COS/S3** – **From a Relational Database** – **To OBS**

Release Date	Description
2018-05-04	This is the seventh official release. • Added the following sections: – **Link to HBase** – **Link to Hive** – **To DDS** • Modified the following sections: – **Step 1: Creating a Cluster** – **Step 3: Creating and Executing a Job** – **Creating a CDM Cluster** – **Creating Links** – **Table/File Migration Jobs** – **Entire DB Migration** – **From a Relational Database** – **Managing a Single Job** • Updated the screenshots. • Changed Elasticsearch Service (ES) to Cloud Search Service (CSS). • Changed Unlimited Query Service (UQuery) to Data Lake Insight (DLI). • Changed Data Pipeline Service (DPS) to Data Lake Factory (DLF).
2018-04-09	This is the sixth official release. • Added the following sections: – **Link to OSS on Alibaba Cloud** – **Link to DLI** – **To DLI** – **From OBS to DLI** • Modified the following sections: – **Step 1: Creating a Cluster** – **Creating a CDM Cluster** – **Modifying Cluster Configurations** – **From OBS/OSS/KODO/COS/S3** – **From HBase/CloudTable** – **Field Conversion** • Modified the procedure for binding EIPs because the EIPs are not automatically bound. • Updated the screenshots.

Release Date	Description
2018-01-31	This is the fifth official release. • Added the following sections: – **From Elasticsearch or CSS** – **Regular Expressions for Separating Semi-structured Text** • Added the JS expression example in **Field Conversion**. • Modified job parameters. • Added the description of selecting a connector in the first step in the procedure for creating a link. • Deleted the following sections: – From VoltDB – To VoltDB – Using CDM to Archive MySQL Data to OBS – Creating the PostgreSQL Link on RDS on HUAWEI CLOUD
2018-01-11	This is the fourth official release. • Added the following sections: – **Link to HDFS** – **Link to CloudTable** – **Link to Kafka** – **Entire DB Migration** – **From Apache Kafka/DMS Kafka** – **From Oracle to CSS** • Modified several connector parameters, job parameters, and corresponding parameter descriptions. • Modified "Procedure" in **Step 3: Creating and Executing a Job**.
2017-11-30	This is the third official release. • Added the following sections: – **Binding or Unbinding an EIP** – **Link to NAS/SFS** – **Link to DIS** – **Link to Elasticsearch/CSS** – **From DIS** – **To Elasticsearch or CSS** – **Field Conversion** • Changed all connector names by deleting **connector** from the names in the document. • Modified content in **Scheduling Job Execution**.

Release Date	Description
2017-10-31	This is the second official release. • Added **Link to MongoDB**. • Added **Scheduling Job Execution**. • Added **Incremental Synchronization Using the Macro Variables of Date and Time**. • Modified the parameter description of the source job configuration and destination job configuration, and enabled the directory, table name, and Where clause to be configured as time macro variables. • Modified the data source list supported by CDM, added the MongoDB data source, and added several data migration scenarios.
2017-09-30	This is the first official release.