

Cloud Data Migration

Getting Started

Issue	01
Date	2023-07-14



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2023. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Contents

1 Overview..... 1

2 Step 1: Creating a Cluster..... 2

3 Step 2: Creating Links..... 5

4 Step 3: Creating and Executing a Job..... 11

5 Step 4: Viewing Job Execution Result..... 14

6 Practices..... 16

1 Overview

This section describes how to use CDM to migrate the data in RDS for MySQL to GaussDB(DWS), thereby helping you get familiar with CDM. [Figure 1-1](#) shows the specific scenario.

Figure 1-1 Migrating data from MySQL to GaussDB(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 Step 1: Creating a Cluster

Scenario

This section describes how to create a CDM cluster to synchronize MySQL data to DWS.

NOTE

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

Note: If a VPC peering connection is created, the peer VPC subnet may overlap with the CDM management network. As a result, data sources in the peer VPC cannot be accessed. You are advised to use the Internet for cross-VPC data migration, or contact the administrator to add specific routes for the VPC peering connection in the CDM background.

- If the CDM cluster and a cloud service are located in different regions, you need to use the Internet or Direct Connect to enable them to communicate with each other. When using the Internet, ensure that an EIP has been bound to the CDM cluster, the security group of CDM allows outbound traffic from the host where the off-cloud data source is located, the host where the data source is located can access the Internet, and the connection port has been enabled in the firewall rules.
- In addition, an enterprise project may also affect the communication between the CDM cluster and other cloud services. The CDM cluster can communicate with a cloud service only if they have the same enterprise project.

Prerequisites

- An RDS for MySQL DB instance has been created, and its region, VPC, subnet, security group, and enterprise project (if any) are the same as those of the CDM cluster.
- A DWS cluster has been created, and its region, VPC, subnet, security group, and enterprise project (if any) are the same as those of the CDM cluster.

- If the region, VPC, subnet, and security group of the RDS for MySQL DB instance or DWS cluster are different from those of the CDM cluster, you must configure a network, EIP, or Direct Connect to enable the communication between the RDS for MySQL DB instance or DWS cluster and the CDM cluster.

Procedure

Step 1 Log in to the CDM management console.

Step 2 Click **buy CDM Cluster** and set the following parameters:

- **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 MySQL instance or DWS 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 any AZ.
- **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**.
- **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 16 jobs can be executed concurrently.
 - **cdm.xlarge:** 16 vCPUs and 32 GB of memory. The maximum and assured bandwidths are 10 Gbit/s and 4 Gbit/s. Up to 32 jobs can be executed concurrently. This flavor is suitable for migrating terabytes of data that requires a bandwidth of 10GE.
 - **cdm.4xlarge:** 64 vCPUs and 128 GB of memory. The maximum and assured bandwidths are 40 Gbit/s and 36 Gbit/s. Up to 128 jobs can be executed concurrently.
- **VPC:** Select the VPC for the RDS for MySQL DB instance or DWS cluster.
- **Subnet:** You are advised to select the subnet of the RDS for MySQL DB instance or DWS cluster.
- **Security Group:** You are advised to select the security group of the RDS for MySQL DB instance or DWS cluster.
- **Enterprise Project:** This parameter is available only when an enterprise project has been created. Select the enterprise project of the RDS for MySQL DB instance or DWS cluster.
- 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.

----End

3

Step 2: Creating Links

Scenario

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

- MySQL link: used to connect to an RDS for MySQL DB instance
- DWS link: used to connect to a DWS cluster

Prerequisites

- You have an RDS for MySQL DB instance and have obtained the name, username, and password for accessing the MySQL database. In addition, you have the read, write, and delete permissions for the MySQL database.
- You have a DWS cluster and have obtained the name, username, and password for accessing the DWS database. In addition, you have the read, write, and delete permissions for the DWS database.
- You have uploaded the MySQL database driver by referring to [Managing Drivers](#).

Creating a MySQL Link

- Step 1** In the left navigation pane, choose **Cluster Management**. 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. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 3-1 Selecting a connector

Data Warehouse	Data Warehouse Service	Data Lake Insight	MRS ClickHouse	
Hadoop	MRS HDFS	Apache HDFS	MRS HBase	Apache HBase
	MRS Hive	Apache Hive	MRS Hudi	
Object Storage	Object Storage Service (OBS)			
File System	FTP	SFTP	HTTP	
Relational Database	RDS for MySQL	MySQL	RDS for PostgreSQL	PostgreSQL
	RDS for SQL Server	Microsoft SQL Server	Oracle	
NoSQL	Redis	MongoDB		
Messaging System	Data Ingestion Service	MRS Kafka	Apache Kafka	loghub
Search	Elasticsearch			
Open Beta Test	<input type="button" value="^"/>			
	<input type="button" value="X Cancel"/>	<input type="button" value=" > Next"/>		

Step 3 Select **RDS for MySQL** and click **Next** to set the link parameters.

Figure 3-2 Creating a MySQL link

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

*

Name

mysqllink

*

Connector

Relational Database

Database Type

MySQL

*

Database Server

*

Port

3306

*

Database Name

sqoop

*

Username

admin

*

Password

Use Local API

Yes

No

Use Agent

Yes

No

Reference Sign

,

local_infile character set

utf8

Driver Version

No matching driver Upload | Copy from SFTP

Show Advanced Attributes

X Cancel

< Previous

Test

Save

Click **Show Advanced Attributes** to view more optional parameters. For details, see [Link to an RDS for MySQL/MySQL Database](#). Retain the default values of the optional parameters and configure the mandatory parameters according to [Table 3-1](#).

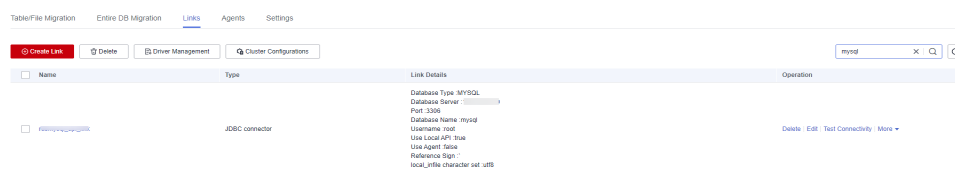
Table 3-1 MySQL link parameters

Parameter	Description	Example Value
Name	Unique link name	mysqllink
Database Server	IP address or domain name of the MySQL database server	-

Parameter	Description	Example Value
Port	MySQL database port	3306
Database Name	Name of the MySQL database	sqoop
Username	User who has the read, write, and delete permissions on the MySQL database	admin
Password	Password of the user	-
Use Local API	Whether to use the local API of the database for acceleration. (The system attempts to enable the local_infile system variable of the MySQL database.)	Yes
Use Agent	Whether to extract data from the data source through an agent	No
local_infile Character Set	When using local_infile to import data to MySQL, you can configure the encoding format.	utf8
Driver Version	Before connecting CDM to a relational database, you need to upload the JDK 8 .jar driver of the relational database. Download the MySQL driver 5.1.48 from https://downloads.mysql.com/archives/c-j/ , obtain mysql-connector-java-5.1.48.jar , and upload it.	-

Step 4 Click **Test** to check whether the parameters are correctly configured. If the test is successful, click **Save** to create the link and return to the **Links** page.

Figure 3-3 Created MySQL link



----End

Creating a DWS Link

Step 1 Click **Job Management** in the **Operation** column of the CDM cluster. On the displayed page, click the **Links** tab and then **Create Link**. The **Select Connector** page is displayed.

Figure 3-4 Selecting a connector

Data Warehouse	Data Warehouse Service	Data Lake Insight	MRS ClickHouse	
Hadoop	MRS HDFS	Apache HDFS	MRS HBase	Apache HBase
	MRS Hive	Apache Hive	MRS Hudi	
Object Storage	Object Storage Service (OBS)			
File System	FTP	SFTP	HTTP	
Relational Database	RDS for MySQL	MySQL	RDS for PostgreSQL	PostgreSQL
	RDS for SQL Server	Microsoft SQL Server	Oracle	
NoSQL	Redis	MongoDB		
Messaging System	Data Ingestion Service	MRS Kafka	Apache Kafka	
Search	Elasticsearch			
Open Beta Test	<input type="checkbox"/>			
	<input type="button" value="X Cancel"/>	<input type="button" value="Next >"/>		

Step 2 Select **Data Warehouse Service** and click **Next** to configure parameters for the DWS link.

Figure 3-5 Creating a DWS link

★ Name

★ Connector

Relational Database

Database Type

Data warehouse

★ Database Server ?

Select

★ Port ?

★ Database Name ?

★ Username ?

★ Password ?

Use Agent ?

Yes

No

Reference Sign ?

Show Advanced Attributes

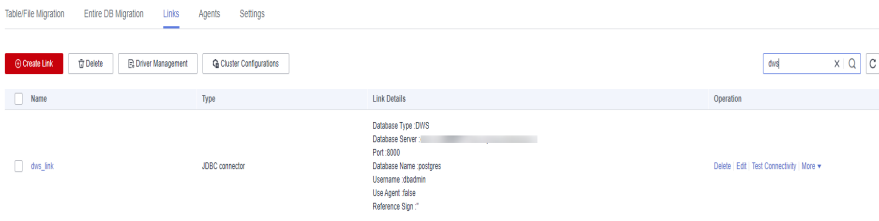
Click **Show Advanced Attributes** and set optional parameters. For details, see [Link to Relational Database](#). **Table 3-2** lists the mandatory parameters. Retain the default values of optional parameters.

Table 3-2 DWS link parameters

Parameter	Description	Example Value
Name	Unique link name	dwslink
Database Server	IP address or domain name of the DWS database server	-
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	No

Step 3 Click **Test** to check whether the parameters are correctly configured. If the test is successful, click **Save** to create the link and return to the **Links** page.

Figure 3-6 Created DWS link



-----End

4

Step 3: Creating and Executing a Job

Scenario

This section describes how to create a job to migrate tables from a 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 4-1 Creating a job

The screenshot displays the 'Create Job' configuration interface. At the top, the 'Job Name' is set to 'mysql2dws'. Below this, the configuration is divided into two main sections: 'Source Job Configuration' and 'Destination Job Configuration'.

Source Job Configuration:

- Source Link Name:** A dropdown menu with 'mysql_link' selected.
- Use SQL Statement:** A toggle switch with 'Yes' selected.
- Schema/Table Space:** A text input field.
- Table Name:** A text input field.
- A link labeled 'Show Advanced Attributes' is visible at the bottom.

Destination Job Configuration:

- Destination Link Name:** A dropdown menu with 'dws_link' selected.
- Schema/Table Space:** A text input field.
- Auto Table Creation:** A dropdown menu with 'Non-auto Creation' selected.
- Table Name:** A text input field.
- Clear Data Before Import:** A dropdown menu with 'Do not clear' selected.
- Import Mode:** A dropdown menu with 'COPY' selected.
- A link labeled 'Hide Advanced Attributes' is visible above the bottom section.
- Is middle Relation table:** A toggle switch with 'Yes' selected.
- PreSql:** A text input field.
- PostSql:** A text input field.
- Number of loader Thread:** A text input field with the value '1'.

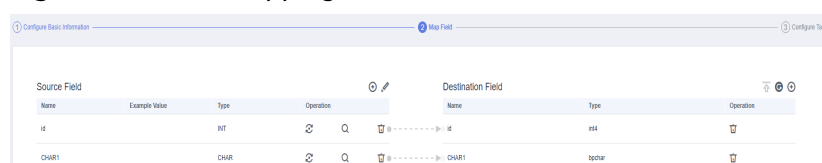
- **Job Name:** Enter a unique job name, for example, **mysql2dws**.
- **Source Job Configuration**
 - **Source Link Name:** Select the **mysqllink** link created in [Step 2: Creating Links](#).

- **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 for other optional parameters. For details, see [From MySQL](#).
- **Destination Job Configuration**
 - **Destination Link Name:** Select the **dwslink** link created in [Step 2: Creating Links](#).
 - **Schema/Tablespace:** Select the schema 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.
 - **Advanced Attributes > Extend Field Length:** Select **Yes**. 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. If this parameter is set to **Yes**, the length of the fields of the character type will be set to three times of its original length when a table is automatically created. This prevents errors caused by insufficient lengths of the characters in DWS tables.
 - Retain the default values for other optional parameters. For details, see [To DWS](#).

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 want to convert the content of the source fields, perform the operations in this step. For details, see [Converting Fields](#). In this example, field conversion is not required.

Figure 4-2 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 If Failed:** Determine whether to automatically retry the job if it fails. Retain the default value **Never**.
- **Group:** Select the group to which the job belongs. The default group is **DEFAULT**. On the **Job Management** page, jobs can be displayed, started, or exported by group.
- **Schedule Execution:** Determine whether to automatically execute the job at a scheduled time. Retain the default value **No** in this example.
- **Concurrent Extractors:** Enter the number of concurrent extractors. An appropriate value improves migration efficiency. For details, see [Performance Tuning](#). Retain the default value **1**.
- **Write Dirty Data:** Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS for future viewing. Before writing dirty data, create an OBS link on the CDM console. Retain the default value **No** so that dirty data is not recorded.

Figure 4-3 Configuring the task

Configure Task

Retry if failed ?

Never

Group ?

DEFAULT

[Add](#) [Edit](#) [Delete](#)

Schedule Execution

YesNo

Hide Advanced Attributes

Concurrent Extractors ?

1

Number of split retries ?

0

Write Dirty Data ?

YesNo

Throttling ?

YesNo

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

Figure 4-4 Job execution

TableFile Migration										Entire DB Migration										Links										Agents										Settings									
<div><div>Create Job</div><div>Run</div><div>Delete</div></div>										<div><div>Export</div><div>Import</div><div>Schedule</div><div>All statuses</div><div>Job namemysqlCdmX</div><div>Q</div><div>C</div></div>																																							
<div><div><div>Enter a group name</div><div>Q</div></div></div>										<input type="checkbox"/>	Name	Link Details		Created By	Execution End Time	Duration	Write Statistics	Status	Group Name	Operation																													
<input type="checkbox"/>	rdmysqlCdm	rdmysqlCdm-link-dms-link			Dec 07, 2022 20:24:07 GMT+08:00	0s	Written rows: 3	Succeeded	DEFAULT	Run	Historical Record	Edit	More																																				

----End

5

Step 4: Viewing Job Execution Result

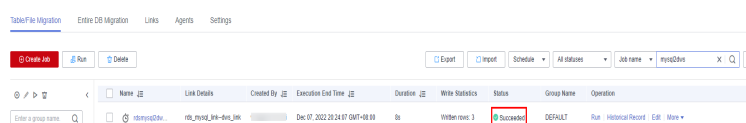
Scenario

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. If the job status is **Succeeded**, the migration is successful.

Figure 5-1 Job status



The screenshot shows the 'Job Management' page with a table of jobs. The job 'mysql_dws' is highlighted with a red box around its status 'Succeeded'.

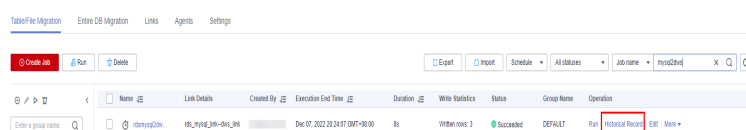
Name	Link Details	Created By	Execution End Time	Duration	Write Statistics	Status	Group Name	Operation
mysql_dws	mysql_dws-link	Dec 07, 2022 20:24:07 GMT+08:00	0s	Written rows: 3	Succeeded	DEFAULT	Run	Historical Record Edit More

NOTE

The job status can be **New**, **Pending**, **Booting**, **Running**, **Failed**, **Succeeded**, or **Stopped**. **Pending** indicates that the job is waiting to be scheduled by the system, and **Booting** indicates that the data to be migrated is being analyzed.

- 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 5-2 Viewing the historical records

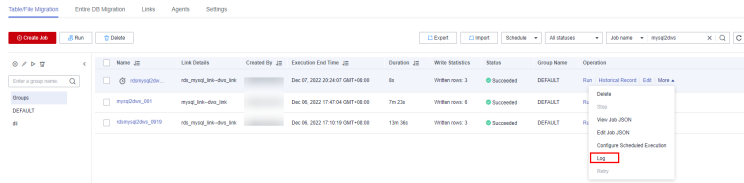


The screenshot shows the 'Historical Record' page for the 'mysql_dws' job. The 'Historical Record' link is highlighted with a red box.

Name	Link Details	Created By	Execution End Time	Duration	Write Statistics	Status	Group Name	Operation
mysql_dws	mysql_dws-link	Dec 07, 2022 20:24:07 GMT+08:00	0s	Written rows: 3	Succeeded	DEFAULT	Run	Historical Record Edit More

- 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 5-3 Viewing job logs



----End

6 Practices

After performing the operations in [Creating a Cluster](#) and [Creating Links](#), you can use a series of common practices provided by CDM as you need.

Table 6-1 Common best practices

Practice		Description
Tutorials	Creating an MRS Hive Link	MRS Hive links are applicable to the MapReduce Service (MRS). This best practice describes how to create an MRS Hive link.
	Migrating Data from MySQL to OBS	CDM supports migration of tables to OBS. This best practice describes how to migrate tables from a MySQL database to OBS.
Parameter transfer	Scheduling a CDM Job by Transferring Parameters Using DataArts Factory	You can use EL expressions in DataArts Factory to transfer parameters to a CDM job to schedule it. This best practice describes how to schedule CDM jobs by transferring parameters in DataArts Factory.
Incremental migration	Incremental File Migration	CDM supports incremental migration of file systems. After a full migration is complete, you can run a job again to export all new files or specified directories or files.
	Incremental Migration of Relational Databases	CDM supports incremental migration of relational databases. After a full migration is complete, you can migrate the incremental data generated within a specified period of time. For example, you can export the data generated on the previous day at 00:00 every day.

Practice		Description
Case study	Case: Trade Data Statistics and Analysis	Consulting company H uses CDM to import local trade statistics to OBS, and Data Lake Insight (DLI) to analyze trade statistics. In this way, company H builds its big data analytics platform at an extremely low cost, allowing the company more time to focus on their businesses and make innovations continuously.
	Case: IoV Big Data Service Migration to Cloud	Company H intends to build an enterprise-class cloud management platform for its IoV service to centrally manage and deploy hardware resources and common software resources, and implement cloud-based and service-oriented transformation of IT applications. Cloud Data Migration (CDM) helps company H build the platform without code modification and data loss.