ALIBABA CLOUD

Alibaba Cloud

DataWorks

Data Integration

Document Version: 20220712

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
⚠ Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	⚠ **Danger:** Resetting will result in the loss of user configuration data.
🔔 Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	🔔 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
🔈 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	🔈 **Notice:** If the weight is set to 0, the server no longer receives new requests.
? Note	A note indicates supplemental instructions, best practices, tips, and other content.	? **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings> Network> Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Overview

Data Integration is a stable, efficient, and scalable data synchronization service. It is designed to migrate and synchronize data between various heterogeneous data sources in complex network environments at a high speed and in a stable manner.

Billing

When you run nodes in Data Integration, you are charged the following fees:

- Fees that are included in your DataWorks bill
  - Fees for using exclusive resource groups for Data Integration or the shared resource group for Data Integration. The shared resource group for Data Integration is used only for debugging.
  - Fees for using exclusive resource groups for scheduling or the shared resource group for scheduling.
  - Fees for the Internet traffic that is generated if data is transmitted over the Internet.
  - Fees for the DataWorks edition that you use.

  > ? **Note** These fees are included in your DataWorks bill.

- Fees that are not included in your DataWorks bill

  You may be charged other fees for the configurations of data synchronization nodes. For example, you may be charged the fees for using data sources, computing and storage features of compute engine instances, and network services such as Express Connect, Elastic IP Address (EIP), and EIP Bandwidth Plan in your data synchronization nodes. These fees are not charged by DataWorks. The bills for these fees are not generated in DataWorks. After you configure and run a data synchronization node, take note of the tasks and fees that are generated when you use the resources of other services.

> ? **Note** For more information about the billable items of DataWorks, see Billing overview.

Limits

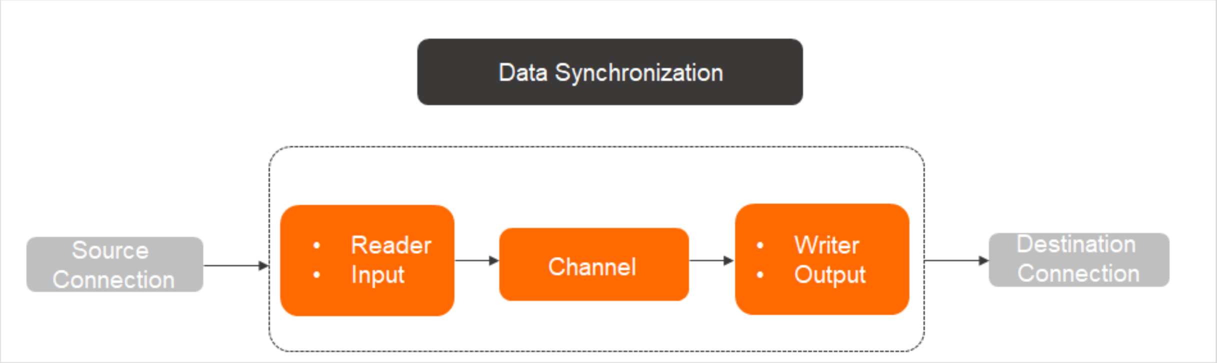
- Data Integration can synchronize structured, semi-structured, and unstructured data. Structured data sources include ApsaraDB RDS and PolarDB-X 1.0. Unstructured data, such as data in Object Storage Service (OSS) objects and text files, must be converted to structured data. Data Integration can synchronize only the data that can be abstracted to two-dimensional logical tables to MaxCompute. Data Integration cannot synchronize unstructured data that cannot be converted to structured data, such as data in MP3 files that are stored in OSS, to MaxCompute.
- Data Integration supports data synchronization and exchange in the same region or across specific regions.

  Data can be transmitted between specific regions over the classic network, but network connectivity cannot be ensured. If the transmission over the classic network fails, we recommend that you transmit data over the Internet.

- Data Integration supports only data synchronization but not data consumption.
- Data synchronization by using Data Integration supports only the at-least-once delivery mechanism. It does not support the exact-once delivery mechanism. This indicates that data synchronized to a destination may be duplicated. You can use a primary key and the capabilities of the destination to ensure the uniqueness of the synchronized data.

Batch synchronization

Data Integration can be used to synchronize large amounts of offline data. Data Integration facilitates data transmission between diverse structured and semi-structured data sources. It provides readers and writers for the supported data sources and defines a transmission channel between sources and destinations based on simplified data types.



Development modes of synchronization nodes

You can develop synchronization nodes in one of the following modes:

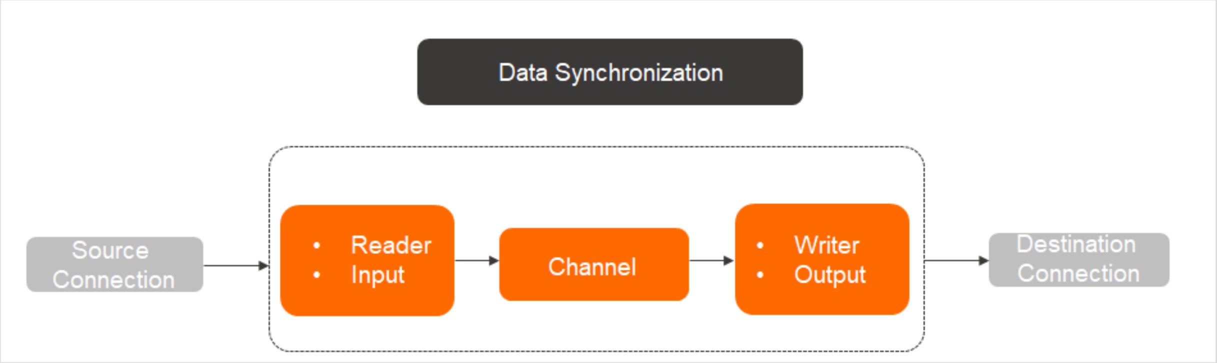
- **Codeless user interface (UI)**: Data Integration provides step-by-step instructions to help you configure a synchronization node. This mode is easy to use but provides only limited features. For more information, see Configure a synchronization node by using the codeless UI.
- **Code editor**: You can write a JSON script to create a synchronization node. This mode supports advanced features to facilitate flexible and fine-grained configuration. It is suitable for experienced users and increases the cost of learning. For more information, see Create a synchronization node by using the code editor.

Note

- The code that is generated for a synchronization node on the codeless UI can be converted to a script. This conversion is irreversible. After the conversion is complete, you cannot switch back to the codeless UI mode.
- Before you write code, you must add data sources to DataWorks and create tables in the destination that are used to store the synchronized data.

Network connectivity

You can run a synchronization node on a resource group for Data Integration to synchronize data from a source to a destination. Before you run the synchronization node, make sure that the resource group for Data Integration is connected to the data sources.



Data Integration allows you to synchronize data between heterogeneous data sources in various network environments. You can select the network connection solution based on the network environment in which the data sources reside to ensure the network connectivity between the resource group for Data Integration and the data sources. For more information, see Select a network connectivity solution.

Data Integration supports data sources that reside on the classic network, in virtual private clouds (VPCs), or in data centers.

- Classic network: a network that is deployed and managed by Alibaba Cloud. The classic network is shared by Alibaba Cloud accounts.
- VPC: a network that is created on Alibaba Cloud and provides an isolated network environment. You have full control over

your VPC. For example, you can customize the IP address range, divide the VPC into multiple subnets, and configure route tables and gateways.

A VPC is an isolated network for which you can specify custom values for parameters, such as the parameters for IP address range, subnets, and gateways. Data Integration provides the feature that automatically detects the reverse proxy for the following data sources based on the wide deployment of VPCs: ApsaraDB RDS for MySQL, ApsaraDB RDS for PostgreSQL, ApsaraDB RDS for SQL Server, PolarDB, PolarDB-X 1.0, HybridDB for MySQL, AnalyticDB for PostgreSQL, and AnalyticDB for MySQL 3.0. This feature frees you from purchasing an Elastic Compute Service (ECS) instance in your VPC to configure synchronization nodes for these data sources. Data Integration uses this feature to automatically detect and establish network connections between these data sources.

When you configure synchronization nodes for other Alibaba Cloud data sources in a VPC, such as ApsaraDB RDS for PPAS, ApsaraDB for OceanBase, ApsaraDB for Redis, ApsaraDB for MongoDB, ApsaraDB for Memcache, Tablestore, and ApsaraDB for HBase data sources, you must purchase an ECS instance in the same VPC. This ECS instance is used to connect to the data sources.

- Data center: a network that is deployed by yourself. This type of network is isolated from Alibaba Cloud networks.

For more information about the classic network and VPCs, see VPC FAQ.

Note You can connect to data sources over the Internet. However, the connection speed depends on the Internet bandwidth, and additional network connection expenses are required. We recommend that you do not connect to data sources over the Internet. For more information about the billing rules of Internet traffic generated during data synchronization, see Billing of Internet traffic.

Terms

- parallelism

  Parallelism indicates the maximum number of parallel threads that a synchronization node uses to read data from a source or write data to a destination.

- bandwidth throttling

  Bandwidth throttling indicates that a maximum transmission rate is specified for a synchronization node in Data Integration.

- dirty data

  Dirty data indicates data that is meaningless to business, does not match the specified data type, or leads to an exception during data synchronization. If an exception occurs when a single data record is written to the destination, the data record is considered as dirty data. Data records that fail to be written to a destination are considered as dirty data. In most cases, dirty data is the data that does not match the specified data type. For example, you want to write VARCHAR-type data in a source to an INT-type field in a destination. A data conversion error occurs, and the data cannot be written to the destination. In this case, the data is dirty data.

  Dirty data cannot be written to a destination. When you configure a data synchronization node, you can specify whether dirty data can be generated. You can also specify the maximum number of dirty data records that can be generated during data synchronization. If the number of generated dirty data records exceeds the upper limit that you specify, the synchronization node fails.

- data source

  A data source is a source from which data is processed by DataWorks. A data source can be a database or a data warehouse. DataWorks supports various types of data sources and data type conversion during data synchronization.

  Before you create a data synchronization node, you can add a source and a destination that you need to use on the **Data Source** page of the DataWorks console. When you create a synchronization node, you must select the source and the destination that you added.

References

- For more information about how to configure a data synchronization node, see Configure a synchronization node by using the codeless UI.
- For more information about how to process unstructured data, such as data stored in OSS objects, see Access OSS data by using a built-in extractor.
- DataWorks provides the shared resource group for Data Integration for you to migrate large amounts of data to the cloud. However, the shared resource group does not work if a high transmission speed is required or your data sources are

deployed in complex network environments. You can use exclusive or custom resource groups for Data Integration to run your data synchronization nodes. This ensures network connections between your data sources and resource groups and ensures a higher transmission speed. For more information, see Create and use an exclusive resource group for Data Integration and Create and use a custom resource group for Data Integration.

2.Batch data synchronization

2.1. Supported data source types, readers, and writers

Data Integration is a stable, efficient, and scalable data synchronization service. It provides transmission channels for offline data that is stored in Alibaba Cloud services such as MaxCompute, AnalyticDB for PostgreSQL, and Hologres.

Notice

- Cross-account data synchronization is supported if the connectivity test is successful. For example, data in a MySQL database that belongs to Account A can be synchronized to a MongoDB database that belongs to Account B.
- Some data sources can be configured only by using the code editor. For more information, see the topics related to readers and writers.
- Excel files cannot be imported. If you want to import an Excel file, convert the Excel file to a CSV file.
- You can synchronize data from or to data sources only after the data sources pass the connectivity test. For more information, see Select a network connectivity solution.

Data source type	Reader	Writer
AWS S3	Amazon S3 Reader	Not supported
AnalyticDB for MySQL 2.0	AnalyticDB for MySQL 2.0 Reader	AnalyticDB for MySQL 2.0 Writer
AnalyticDB for MySQL 3.0	AnalyticDB for MySQL 3.0 Reader	AnalyticDB for MySQL 3.0 Writer
AnalyticDB for PostgreSQL	AnalyticDB for PostgreSQL Reader	AnalyticDB for PostgreSQL Writer
ApsaraDB For Oceanbase Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	ApsaraDB For Oceanbase Reader	ApsaraDB For Oceanbase Writer
ClickHouse Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	ClickHouse Reader	ClickHouse Writer
DataHub	DataHub Reader	DataHub Writer

Data source type	Reader	Writer
DB2 Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	DB2 Reader	DB2 Writer
Dameng (DM) Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	DM Reader	DB Writer
DRDS	DRDS Reader	DRDS Writer
Elasticsearch	Elasticsearch Reader	Elasticsearch Writer
FTP	FTP Reader	FTP Writer
GBase8a Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	Supported	Supported
HBase Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	• HBase Reader • HBase20xsql Reader	• HBase Writer • HBase 11xsql Writer
HDFS Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	HDFS Reader	HDFS Writer

Data source type	Reader	Writer
Hive Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	Hive Reader	Hive Writer
Hologres Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	Supported	Supported
HybridDB for MySQL	HybridDB for MySQL Reader	HybridDB for MySQL Writer
Kafka Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	Kafka Reader	Kafka Writer
KingbaseES Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	KingbaseES Reader	KingbaseES Writer
Lindorm Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	Lindorm Reader	Lindorm Writer
LogHub (SLS)	LogHub (SLS) Reader	LogHub (SLS) Writer

Data source type	Reader	Writer
MaxCompute	MaxCompute Reader	MaxCompute Writer
MaxGraph	Not supported	Maxgraph Writer
Memcache	Not supported	Memcache Writer
MetaQ Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	MetaQ Reader	Not supported
MongoDB Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	MongoDB Reader	MongoDB Writer
MySQL Note MySQL 8.0 data sources support only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	MySQL Reader	MySQL Writer
OpenSearch Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	Not supported	OpenSearch Writer
Oracle	Oracle Reader	Oracle Writer
OSS	OSS Reader	OSS Writer
OTSStream	OTSStream Reader	Not supported
PolarDB	PolarDB Reader	PolarDB Writer
PostgreSQL	PostgreSQL Reader	PostgreSQL Writer

Data source type	Reader	Writer
Prometheus Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	Prometheus Reader	Not supported
Redis	Not supported	Redis Writer
HTTP RESTful API Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	RestAPI Reader	RestAPI Writer
SAP HANA Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	SAP HANA Reader	SAP HANA Writer
Stream	Stream Reader	Stream Writer
SQL Server	SQL Server Reader	SQL Server Writer
Tablestore	Tablestore Reader	Tablestore Writer
TSDB Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	Not supported	TSDB Writer
Vertica Note This type of data source supports only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.	Vertica Reader	Vertica Writer

2.2. Node configuration

2.2.1. Configure a synchronization node by using the codeless UI

This topic describes how to configure a synchronization node by using the codeless user interface (UI) in Data Integration.

Procedure

1. Add data sources.
2. Create a batch synchronization node.
3. Select the source.
4. Select the destination.
5. Map the fields in the source and destination tables.
6. Configure channel control policies, such as the maximum transmission rate and the maximum number of dirty data records allowed.
7. Configure the properties of the synchronization node.

Add data sources

A synchronization node can synchronize data between various homogeneous or heterogeneous data sources. On the DataStudio page of the DataWorks console, click the **Workspace Manage** icon in the upper-right corner. On the page that appears, click **Data Source** in the left-side navigation pane. On the Data Source page, add a data source. For more information, see Connection configuration.

After you add a data source, you can select the data source when you configure a synchronization node on the DataStudio page. For more information about the types of data sources that are supported by Data Integration, see Supported data source types, readers, and writers.

> Note
>
> - Data Integration does not support connectivity testing for some types of data sources. For more information, see Select a network connectivity solution.
> - If an on-premises data source does not have a public IP address or is not accessible from a network, the connectivity testing fails when you configure the data source. To resolve the connection failure, you can use a custom resource group to connect to the data source. For more information, see Create and use a custom resource group for Data Integration. If a data source is not accessible from a network, Data Integration cannot obtain the table schemas of the data source. In this case, you can configure a synchronization node for this data source only by using the code editor.

Create a workflow

1. Log on to the DataWorks console.
2. In the left-side navigation pane, click **Workspaces**.
3. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
4. On the **DataStudio** page, move the pointer over the icon and select **Workflow**.
5. In the **Create Workflow** dialog box, specify **Workflow Name** and **Description**.

   > Notice The workflow name must be 1 to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).

6. Click **Create**.

Create a batch synchronization node

1. Click the newly created workflow and right-click **Data Integration**.
2. Choose **Create > Batch Synchronization**.

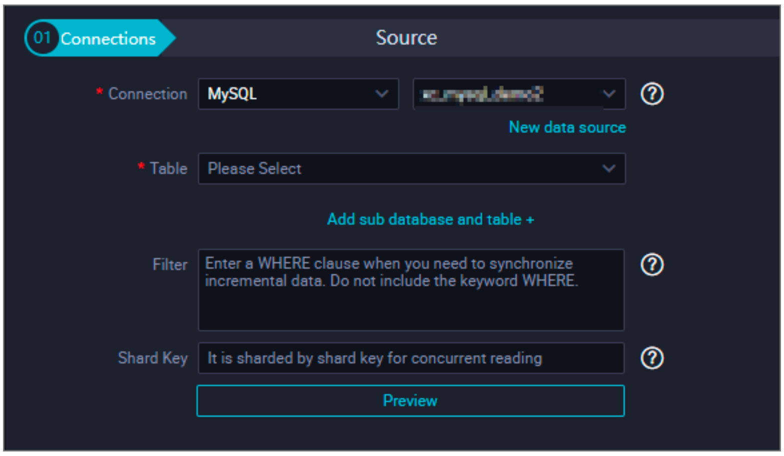
3. In the **Create Node** dialog box, configure the **Node Name** and **Location** parameters.

   Notice The node name must be 1 to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).

4. Click **Commit**.

Select the source

After you create a batch synchronization node, you must select a data source and a table in the Source section.



* Connection: MySQL [redacted]

* Table: Please Select

Filter: Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.

Shard Key: It is sharded by shard key for concurrent reading

Note

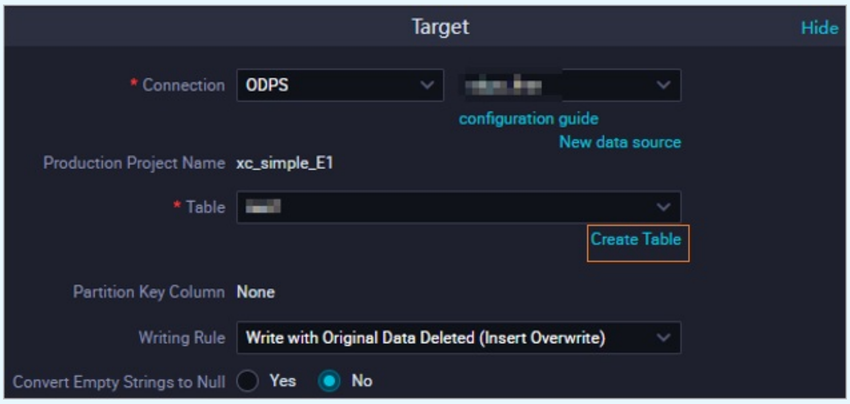
- For more information about how to set the parameters in the Source section, see Configure a reader.
- By default, the Table drop-down list displays a maximum of 25 tables. If the selected data source contains more than 25 tables and the table that you want to select is not displayed in the Table drop-down list, enter the name of the table in the Table field. Alternatively, configure the batch synchronization node by using the code editor.
- Some synchronization nodes may need to synchronize incremental data. In this case, you can use the scheduling parameters of DataWorks to specify the date and time for incremental data synchronization. For more information, see Overview of scheduling parameters.

Select the destination

After you configure the source, you must select a data source and a table in the Target section.

Note

- For more information about how to set the parameters in the Target section, see Configure a writer.
- You must specify a write mode, such as overwriting or appending, for most synchronization nodes. The write mode that you can specify for a synchronization node varies based on the data source type.
- After you select a data source as the destination, you can click **Create Table** below the Table drop-down list. The Create Table dialog box appears and displays the table creation statements. You can modify the table creation statements as needed. The automatic table creation feature is available only for some data sources. Check whether the table creation statements meet your requirements.



Target

* Connection: ODPS [redacted]

Production Project Name: xc_simple_E1

* Table: [redacted]

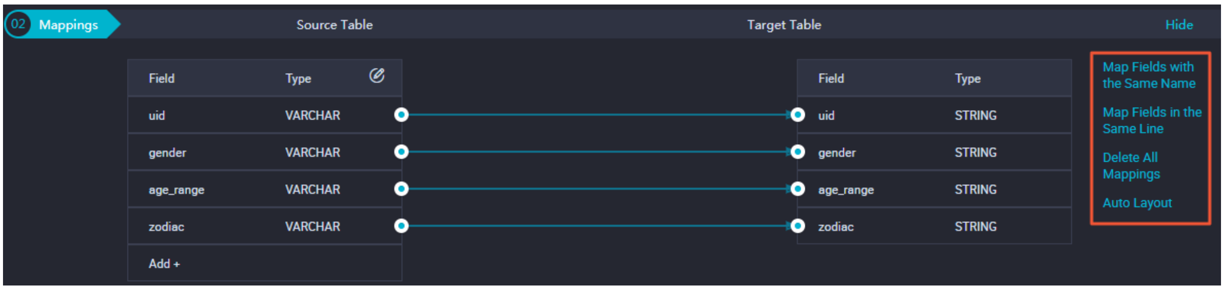
Partition Key Column: None

Writing Rule: Write with Original Data Deleted (Insert Overwrite)

Convert Empty Strings to Null: ○ Yes ● No

Map the fields in the source and destination tables

After you specify the source and destination tables, you must configure the mappings between fields in the source and destination tables. You can click **Map Fields with the Same Name**, **Map Fields in the Same Line**, **Delete All Mappings**, or **Auto Layout** to perform the related operation.



02 Mappings

Source Table

Field	Type
uid	VARCHAR
gender	VARCHAR
age_range	VARCHAR
zodiac	VARCHAR
Add +	

Target Table

Field	Type
uid	STRING
gender	STRING
age_range	STRING
zodiac	STRING

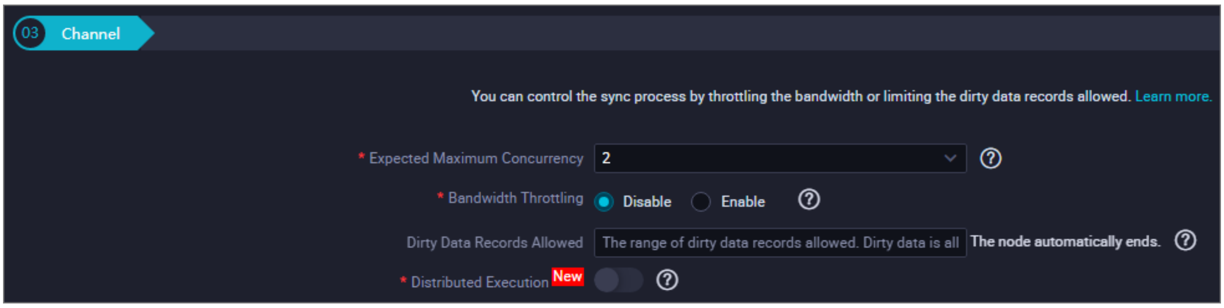
GUI element	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout to sort the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.

GUI element	Description
Add	- You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. - You can use scheduling parameters, such as ${bizdate}. - You can specify the partition key columns from which you want to read data, such as **pt**. - You can use functions supported by relational databases, such as now() and count(1). MaxCompute functions are not supported. - If the field that you entered cannot be parsed, the value of the Type parameter for the field is displayed as **Custom**. For example, if you add a partition key column of a MaxCompute table or a column of a LogHub table that cannot be previewed, the value of the Type parameter for this column is displayed as **Custom**. This does not affect the execution of the synchronization node.

> **Note** Make sure that the data type of a source field is the same as that of the mapped destination field or the data type conversion is feasible.

Configure channel control policies

After you complete the preceding steps, you can configure channel control policies for the synchronization node.



03 Channel

You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	Specifies whether to enable the distributed execution mode. In distributed mode, your synchronization node can be sliced and distributed to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node, excessive access requests are sent to the data sources. Evaluate the access load on the data sources before you use this mode. You can use this mode only when you use exclusive resource groups for Data Integration.

Configure the properties of the synchronization node

In most cases, synchronization nodes use scheduling parameters to filter data. This section describes how to set scheduling parameters for a synchronization node.

On the configuration tab of the batch synchronization node, click **Properties** in the right-side navigation pane.

You can specify a variable in scheduling parameters in the format of ${Variable name}. After a variable is specified, enter the initial value of the variable in the corresponding field. In this example, the initial value of the variable is identified by $[]. The content can be a time expression or a constant.

For example, if you write ${today} in the code and enter today=$[yyyymmdd] in the corresponding field, the value of the time variable is the current date. For more information about how to add or subtract the date, see Overview of scheduling parameters.

In the Properties panel, you can configure the properties of the synchronization node, such as the recurrence, time when the synchronization node is run, and dependencies. Batch synchronization nodes do not have ancestor nodes because they are run before extract, transform, and load (ETL) nodes. We recommend that you specify the root node of the workspace as their ancestor node.

2.2.2. Create a synchronization node by using the code editor

This topic describes how to create a synchronization node by using the code editor.

Development procedure

To create a synchronization node by using the code editor, perform the following steps:

1. Add a data source.
2. Create a batch synchronization node.
3. Apply a template.
4. Configure a reader for the synchronization node.
5. Configure a writer for the synchronization node.
6. Configure field mappings.
7. Configure channel control policies, such as the maximum transmission rate and the maximum number of dirty data records allowed.
8. Configure scheduling properties for the synchronization node.

Add a data source

A synchronization node can synchronize data between various homogeneous or heterogeneous data sources. On the DataStudio page of the DataWorks console, click the **Workspace Manage** icon in the upper-right corner. On the page that appears, click **Data Source** in the left-side navigation pane. On the Data Source page, add a data source. For more information, see Add a data source.

After you add a data source, you can select it when you configure a synchronization node on the DataStudio page. For more information about the types of data sources that are supported by Data Integration, see Supported data source types, readers, and writers.

Note

- Data Integration does not support connectivity testing for some data source types. For more information, see Select a network connectivity solution.
- If an on-premises data source does not have a public IP address or is not accessible from a network, the connectivity testing fails when you configure the data source. You can use a custom resource group to resolve the connection failure. For more information, see Create and use a custom resource group for Data Integration.

  If a data source cannot be directly connected over a network, Data Integration cannot obtain the table schema. In this case, you can create a synchronization node for this data source only by using the code editor.

Create a workflow

1. Log on to the DataWorks console.
2. In the left-side navigation pane, click **Workspaces**.
3. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.

4. On the DataStudio page, move the pointer over the icon and select **Workflow**.
5. In the **Create Workflow** dialog box, specify **Workflow Name** and **Description**.

> **Notice** The workflow name must be 1 to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).

6. Click **Create**.

Create a batch synchronization node

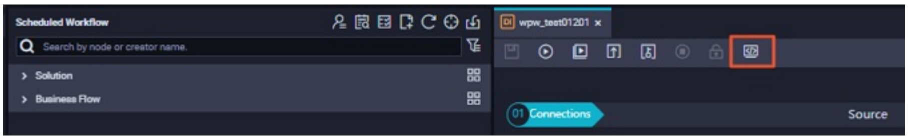
1. Click the newly created workflow and right-click **Data Integration**.
2. Choose **Create > Batch Synchronization**.
3. In the **Create Node** dialog box, configure the **Node Name** and **Location** parameters.

> **Notice** The node name must be 1 to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).

4. Click **Commit**.

Apply a template

1. On the node configuration tab that appears, click the **Switch to Code Editor** icon in the top toolbar.



2. In the **Confirm** dialog box, click **OK**.

> **Note** The code editor supports more features than the codeless user interface (UI). For example, you can configure synchronization nodes in the code editor even when the connectivity test fails.

3. Click the icon in the top toolbar.
4. In the **Apply Template** dialog box, configure the following parameters: **Source Connection Type**, **Connection**, **Target Connection Type**, and **Connection**.
5. Click **OK**.

Configure a reader for the synchronization node

After the template is applied, the basic settings of the reader are generated. You can configure the source and source table based on your business requirements.

```
{"type": "job",
    "version": "2.0",
    "steps": [   // Do not modify the preceding lines. They indicate the header code of the synchronization nod
e.
        {
            "stepType": "mysql",
            "parameter": {
                "datasource": "MySQL",
                "column": [
                    "id",
                    "value",
                    "table"
                ],
                "socketTimeout": 3600000,
                "connection": [
                    {
                        "datasource": "MySQL",
                        "table": [
                            "`case`"
                        ]
                    }
                ],
                "where": "",
                "splitPk": "",
                "encoding": "UTF-8"
            },
            "name": "Reader",
            "category": "reader"    // Specifies that these settings are related to the reader.
        },
```

Parameter description:

- type: the type of the synchronization node. You must set the value to job.
- version: the version number of the synchronization node. You can set the value to 1.0 or 2.0.

Note

- For more information about how to configure the source, see Configure MaxCompute Reader.
- Some synchronization nodes may need to synchronize incremental data. In this case, you can use the scheduling parameters of DataWorks to specify the date and time for incremental data synchronization. For more information, see Overview of scheduling parameters.

Configure a writer for the synchronization node

After the reader is configured, you can configure the destination and destination table based on your business requirements.

```
{
  "stepType": "odps",
  "parameter": {
      "postSql":[], // The SQL statement that you want to execute after the synchronization node is run.
      "partition": "",
      "truncate": true,
      "compress": false,
      "datasource": "odps_first",
      "column": [
          "*"
       ],
       "emptyAsNull": false,
       "table": "",
       "preSql":[
               "delete from XXX;" // The SQL statement that you want to execute before the synchronization node is run. Separate multiple statements with semicolons (;).
             ]
     },
     "name": "Writer",
     "category": "writer"   // Specifies that these settings are related to the writer.
   }
 ],
```

Note

- For more information about how to configure the destination, see Configure MaxCompute Writer.
- You can select the writing method for most nodes. For example, the writing method can be overwriting or appending. Supported writing methods vary based on the data source type.

Map the fields in the source and destination tables

The code editor supports only the mappings of fields in the same row. The data types of the fields must match.

Note Make sure that the data type of a source field is the same as that of the mapped destination field or the data type conversion is feasible.

Configure channel control policies

After the preceding steps are performed, you can configure the channel control policies for the synchronization node. The setting parameter specifies node efficiency parameters, including the number of parallel threads, bandwidth throttling, dirty data policy, and resource group.

```
"setting": {
        "errorLimit": {
            "record": "1024"   // The maximum number of dirty data records allowed.
        },
        "speed": {
            "throttle": false,   // Specifies whether to enable bandwidth throttling.
            "concurrent": 1 // The maximum number of parallel threads.
        }
    },
```

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI. For example, if you set the concurrent parameter to 8 and you want to read data from the same table in two instances, the synchronization node uses a maximum of eight threads to read data from or write data to the two instances in parallel. The eight parallel threads are randomly allocated to the two instances.

Parameter	Description
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.

Configure scheduling properties of the synchronization node

In most cases, synchronization nodes use scheduling parameters to filter data. This section describes how to configure scheduling parameters for a synchronization node.

On the **DataStudio** page, double-click the batch synchronization node in the related workflow. On the node configuration tab, click the **Properties** panel in the right-side navigation pane to configure scheduling properties for the node.

In the Properties panel, you can configure the scheduling properties of the synchronization node, such as the recurrence, time when the synchronization node is run, and dependencies. Batch synchronization nodes do not have ancestor nodes because they are run before extract, transform, and load (ETL) nodes. We recommend that you specify the root node of the workspace as their ancestor node.

After the synchronization node is configured, save and commit the node. For more information about the node scheduling properties, see Basic properties.

2.2.3. Use API operations to create, configure, and manage a data synchronization node

This topic describes how to use API operations to create, configure, and manage a data synchronization node in Data Integration.

Prerequisites

- A Maven project is created. For more information, see Create a Maven project.
- A workflow is created. For more information, see Create a workflow.
- Data sources are created and added to DataWorks for data synchronization. For more information, see Add a data source.

Limits

- In DataWorks, you can call the CreateDISyncTask API operation to create only batch data synchronization nodes.
- You can use only the code editor to configure data synchronization nodes that are created by calling the CreateDISyncTask API operation. For more information, see Create a synchronization node by using the code editor.
- In DataWorks, you cannot use an API operation to create a workflow. You can call the CreateDISyncTask API operation to create data synchronization nodes only in existing workflows.

Preparations

- Configure Maven dependencies.
  i. Open the *pom.xml* file of your Maven project and add `aliyun-java-sdk-core` to the file.

```
 <dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>4.5.20</version>
</dependency>
```

  ii. Open the *pom.xml* file of your Maven project and add `aliyun-java-sdk-dataworks-public` to the file.

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-dataworks-public</artifactId>
  <version>3.4.2</version>
</dependency>
```

- Authenticate an account.

  Before you can use an API operation to create a data synchronization node, you must run the following code to authenticate the Alibaba Cloud account that you want to use to log on to DataWorks. If the account passes the authentication, you can perform subsequent operations. If the account fails the authentication, an error is returned, and you must resolve the issue based on the error.

  ```
   DefaultProfile profile = DefaultProfile.getProfile(
                            "regionId",            // The ID of the region where your DataWorks workspace resi
  des, such as cn-hangzhou.
                            "<yourAccessKeyId>",   // The AccessKey ID of the Alibaba Cloud account that you w
  ant to use to access the DataWorks workspace.
                            "<yourAccessSecret>"); // The AccessKey secret of the Alibaba Cloud account that y
  ou want to use to access the DataWorks workspace.
   IAcsClient client = new DefaultAcsClient(profile);
  ```

  To obtain the AccessKey ID and AccessKey secret of your Alibaba Cloud account, log on to the DataWorks console, move the pointer over your profile picture, and then click **AccessKey Management**.

Overview

After you complete the preceding preparations, you can call API operations to perform the following steps:

1. Create a data synchronization node in Data Integration.
2. Configure scheduling dependencies for the node.
3. Commit the node.
4. Query the status of the node.
5. Deploy the node to the production environment.
6. Query the status of the node.

Procedure

1. Create a data synchronization node in Data Integration.

   Call the *CreateDISyncTask* operation to create a data synchronization node in Data Integration. The following code provides an example of the settings of some parameters. For more information, see CreateDISyncTask.

```
public void createFile() throws ClientException{
        CreateDISyncTaskRequest request = new CreateDISyncTaskRequest();
        request.setProjectId(181565L);
        request.setTaskType("DI_OFFLINE");
        request.setTaskContent("{\"type\":\"job\",\"version\":\"2.0\",\"steps\":[{\"stepType\":\"mysql\",\"
parameter\":{\"envType\":1,\"datasource\":\"dh_mysql\",\"column\":[\"id\",\"name\"],\"tableComment\":\"Comm
ent for the table same\",\"connection\":[{\"datasource\":\"dh_mysql\",\"table\":[\"same\"]}],\"where\":\"\"
,\"splitPk\":\"id\",\"encoding\":\"UTF-8\"},\"name\":\"Reader\",\"category\":\"reader\"},{\"stepType\":\"od
ps\",\"parameter\":{\"partition\":\"pt=${bizdate}\",\"truncate\":true,\"datasource\":\"odps_first\",\"envTy
pe\":1,\"column\":[\"id\",\"name\"],\"emptyAsNull\":false,\"tableComment\":\"Comment for the table same\",\
"table\":\"same\"},\"name\":\"Writer\",\"category\":\"writer\"}],\"setting\":{\"errorLimit\":{\"record\":\"
\"},\"speed\":{\"throttle\":false,\"concurrent\":2}},\"order\":{\"hops\":[{\"from\":\"Reader\",\"to\":\"Wri
ter\"}]}}");
        request.setTaskParam("{\"FileFolderPath\":\"Business Flow/new_biz/Data Integration\",\"ResourceGrou
p\":\"S_res_group_280749521950784_1602767279794\"}");
        request.setTaskName("new_di_task_0607_1416");
        String akId = "XXX";
        String akSecret = "XXXX";
        String regionId = "cn-hangzhou";
        IClientProfile profile = DefaultProfile.getProfile(regionId, akId, akSecret);
        DefaultProfile.addEndpoint("cn-hangzhou","dataworks-public","dataworks.cn-hangzhou.aliyuncs.com");
        IAcsClient client;
        client = new DefaultAcsClient(profile);
        CreateDISyncTaskResponse response1 = client.getAcsResponse(request);
        Gson gson1 = new Gson();
        System.out.println(gson1.toJson(response1));
}
```

2. Configure scheduling dependencies for the node.

   Call the *UpdateFile* operation to configure scheduling dependencies for the node.

```
public static void updateFile(Long fileId) throws Exception {
        UpdateFileRequest request = new UpdateFileRequest();
        request.setProjectId(2043L);
        request.setFileId(fileId);
        request.setAutoRerunTimes(3);
        request.setRerunMode("FAILURE_ALLOWED");
        request.setCronExpress("00 30 05 * * ?");
        request.setCycleType("DAY");
        request.setResourceGroupIdentifier("S_res_group_XXX");
        // Use an exclusive resource group for scheduling.
        request.setInputList("dataworks_di_autotest_root");
        UpdateFileResponse response1 = client.getAcsResponse(request);
    }
```

   The preceding code provides an example of the settings of some parameters. For more information, see UpdateFile.

3. Commit the node.

   Call the *SubmitFile* operation to commit the node to the development environment of the scheduling system.

```
 public void submitFile() throws ClientException{
        SubmitFileRequest request = new SubmitFileRequest();
        request.setProjectId(78837L);
        request.setProjectIdentifier("zxy_8221431");
      // The ID is the value of the FileId parameter that is returned when you create the node.
        request.setFileId(501576542L);
        request.setComment("Comment");
        SubmitFileResponse acsResponse = client.getAcsResponse(request);
}
```

   The preceding code provides an example of the settings of some parameters. For more information, see SubmitFile.

4. Query the status of the node.

   After you commit the node, the ID of the deployment task of the node is returned. You can call the GetDeployment operation to query the details about the deployment task based on the ID. If the value of the Status parameter in the response of the GetDeployment operation is *1*, the deployment task is successful, and the node is committed. You can deploy a node only after the node is committed. If your node fails to be committed, you must handle the failure based on the error that is returned.

```
public void getDeployment() throws ClientException{
        GetDeploymentRequest request = new GetDeploymentRequest();
        request.setProjectId(78837L);
        request.setProjectIdentifier("zxy_8221431");
      // The ID is the deployment task ID that is returned after you commit the node.
        request.setDeploymentId(2776067L);
        GetDeploymentResponse acsResponse = client.getAcsResponse(request);
        log.info(acsResponse.getData().toString());
    }
```

The preceding code provides an example of the settings of some parameters. For more information, see GetDeployment.

5. Deploy the node to the production environment.

   Call the DeployFile operation to deploy the node to the production environment.

   > Note If your workspace is in standard mode, you must perform this step.

```
 public void deploy() throws ClientException{
        DeployFileRequest request = new DeployFileRequest();
        request.setProjectIdentifier("zxy_8221431");
        request.setFileId(501576542L);
        request.setComment("Comment");
        // You must specify the NodeId or FileId parameter. The value of NodeId is the value of the Node ID parameter in the General section of the Properties tab on the configuration tab of the node.
        request.setNodeId(700004537241L);
        DeployFileResponse acsResponse = client.getAcsResponse(request);
    }
```

   The preceding code provides an example of some parameter settings. For more information, see DeployFile.

6. Query the status of the node.

   After you deploy the node, the ID of the deployment task of the node is returned. You can call the GetDeployment operation to query the details about the deployment task based on the ID. If the value of the Status parameter in the response of the GetDeployment operation is *1*, the deployment task is successful, and the node is deployed.

```
public void getDeployment() throws ClientException{
        GetDeploymentRequest request = new GetDeploymentRequest();
        request.setProjectId(78837L);
        request.setProjectIdentifier("zxy_8221431");
      // The ID is the deployment task ID that is returned after you deploy the node.
        request.setDeploymentId(2776067L);
        GetDeploymentResponse acsResponse = client.getAcsResponse(request);
        log.info(acsResponse.getData().toString());
    }
```

   The preceding code provides an example of the settings of some parameters. For more information, see GetDeployment.

Manage a data synchronization node

After you create and configure a data synchronization node in Data Integration, you can perform the following operations on the node:

- Modify the configuration of the node: You can call the UpdateDISyncTask operation to modify the configuration of the node. You can also use the TaskParam parameter to change the exclusive resource group that is used by the node. For more information, see UpdateDISyncTask.

  > Note After you modify the configuration of a data synchronization node, you must commit and deploy the node again. For more information, see Overview.

- Backfill data for the node: You can call the RunCycleDagNodes operation to backfill data for the node. This operation allows you to backfill data for multiple nodes in a workflow at the same time. For more information, see

RunCycleDagNodes.

Example

The following example shows the complete code that is used to create and configure a batch data synchronization node in Data Integration:

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.dataworks_public.model.v20200518.*;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;
import java.util.List;
public class createofflineTask {
    static Long createTask(String fileName) throws  Exception {
        Long projectId = 2043L;
        String taskType = "DI_OFFLINE";
        String taskContent = "{\n" +
                "    \"type\": \"job\",\n" +
                "    \"version\": \"2.0\",\n" +
                "    \"steps\": [\n" +
                "        {\n" +
                "            \"stepType\": \"mysql\",\n" +
                "            \"parameter\": {\n" +
                "                \"envType\": 0,\n" +
                "                \"datasource\": \"mysql_autotest_dev\",\n" +
                "                \"column\": [\n" +
                "                    \"id\",\n" +
                "                    \"name\"\n" +
                "                ],\n" +
                "                \"connection\": [\n" +
                "                    {\n" +
                "                        \"datasource\": \"mysql_autotest_dev\",\n" +
                "                        \"table\": [\n" +
                "                            \"user\"\n" +
                "                        ]\n" +
                "                    }\n" +
                "                ],\n" +
                "                \"where\": \"\",\n" +
                "                \"splitPk\": \"\",\n" +
                "                \"encoding\": \"UTF-8\"\n" +
                "            },\n" +
                "            \"name\": \"Reader\",\n" +
                "            \"category\": \"reader\"\n" +
                "        },\n" +
                "        {\n" +
                "            \"stepType\": \"odps\",\n" +
                "            \"parameter\": {\n" +
                "                \"partition\": \"pt=${bizdate}\",\n" +
                "                \"truncate\": true,\n" +
                "                \"datasource\": \"odps_first\",\n" +
                "                \"envType\": 0,\n" +
                "                \"column\": [\n" +
                "                    \"id\",\n" +
                "                    \"name\"\n" +
                "                ],\n" +
                "                \"emptyAsNull\": false,\n" +
                "                \"tableComment\": \"null\",\n" +
                "                \"table\": \"user\"\n" +
                "            },\n" +
                "            \"name\": \"Writer\",\n" +
                "            \"category\": \"writer\"\n" +
                "        }\n" +
                "    ],\n" +
                "    \"setting\": {\n" +
                "        \"executeMode\": null,\n" +
                "        \"executeMode\": null,\n" +
                "        \"errorLimit\": {\n" +
                "            \"record\": \"\"\n" +
                "        },\n" +
                "        \"speed\": {\n" +
                "            \"concurrent\": 2,\n" +
                "            \"throttle\": false\n" +
                "        }\n" +
                "    },\n" +
                "    \"order\": {\n" +
                "        \"hops\": [\n" +
                "            {\n" +
                "                \"from\": \"Reader\",\n" +
                "                \"to\": \"Writer\"\n" +
                "            }\n" +
                "        ]\n" +
                "    }\n" +
                "}";
        CreateDISyncTaskRequest request = new CreateDISyncTaskRequest();
        request.setProjectId(projectId);
        request.setTaskType(taskType);
        request.setTaskContent(taskContent);
        request.setTaskName(fileName);
        request.setTaskParam("{\"FileFolderPath\":\"Business Flow/test/Data Integration\",\"ResourceGroup\":\"S
_res_group_XXX\"}");
        // Use an exclusive resource group for Data Integration.
        CreateDISyncTaskResponse response1 = client.getAcsResponse(request);
        return response1.getData().getFileId();
    }
    public static void updateFile(Long fileId) throws Exception {
        UpdateFileRequest request = new UpdateFileRequest();
        request.setProjectId(2043L);
        request.setFileId(fileId);
        request.setAutoRerunTimes(3);
        request.setRerunMode("FAILURE_ALLOWED");
        request.setCronExpress("00 30 05 * * ?");
        request.setCycleType("DAY");
        request.setResourceGroupIdentifier("S_res_group_XXX");
        // Use an exclusive resource group for scheduling.
        request.setInputList("dataworks_di_autotest_root");
        UpdateFileResponse response1 = client.getAcsResponse(request);
    }
    public static  Long submitFile(Long fileId) throws  Exception {
        SubmitFileRequest request = new SubmitFileRequest();
        request.setProjectId(2043L);
        request.setFileId(fileId);
        SubmitFileResponse acsResponse = client.getAcsResponse(request);
        Long deploymentId = acsResponse.getData();
        return deploymentId;
    }
    public static  void getDeployment(Long deploymentId) throws Exception {
        GetDeploymentRequest request = new GetDeploymentRequest();
        request.setProjectId(2043L);
        request.setDeploymentId(deploymentId);
        GetDeploymentResponse acsResponse = client.getAcsResponse(request);
        System.out.println(acsResponse.getData().getDeployment().getStatus());
    }
    public static  Long deploy(Long fileId) throws Exception {
        DeployFileRequest request = new DeployFileRequest();
        request.setProjectId(2043L);
        request.setFileId(fileId);
        DeployFileResponse acsResponse = client.getAcsResponse(request);
        Long deploymentId = acsResponse.getData();
        return deploymentId;
    }
    public static Long listNode(String nodeName) throws Exception {
```

```
    public static Long listNode(String nodeName) throws Exception {
        ListNodesRequest request = new ListNodesRequest();
        request.setProjectId(2043L);
        request.setNodeName(nodeName);
        request.setProjectEnv("PROD");
        ListNodesResponse acsResponse = client.getAcsResponse(request);
        List<ListNodesResponse.Data.NodesItem> nodesItemList = acsResponse.getData().getNodes();
        return nodesItemList.get(0).getNodeId();
    }
    public static void RunCycleDagNodes(Long nodeId) throws Exception {
        RunCycleDagNodesRequest request = new RunCycleDagNodesRequest();
        request.setIncludeNodeIds(nodeId.toString());
        request.setName("rerun_job");
        request.setParallelism(false);
        request.setProjectEnv("PROD");
        request.setRootNodeId(nodeId);
        request.setStartBizDate("2021-09-29 00:00:00");
        request.setEndBizDate("2021-09-29 00:00:00");
        request.setProjectEnv("PROD");
        RunCycleDagNodesResponse acsResponse = client.getAcsResponse(request);
    }
    static IAcsClient client;
    public static void main(String[] args) throws Exception {
        String akId = "XX";
        String akSecret = "XX"; // Specify the AccessKey secret of the Alibaba Cloud account that you use to access your DataWorks workspace.
        String regionId = "cn-chengdu";
        IClientProfile profile = DefaultProfile.getProfile(regionId, akId, akSecret);
        DefaultProfile.addEndpoint(regionId, "dataworks-public", "dataworks." + regionId + ".aliyuncs.com");
        client = new DefaultAcsClient(profile);
        String taskName = "offline_job_0930_1648";
        Long fileId = createTask(taskName); // Create a data synchronization node in Data Integration.
        updateFile(fileId);   // Configure scheduling properties for the node.
        Long deployId = submitFile(fileId); // Commit the node.
        getDeployment(deployId);  // Query the status of the node.
        Thread.sleep(10000); // Wait until the node is committed.
        getDeployment(deployId);  // Query the status of the node.
        deployId = deploy(fileId);  // Deploy the node to the production environment.
        getDeployment(deployId);    // Query the status of the node.
        Thread.sleep(10000);        // Wait until the node is deployed.
        getDeployment(deployId);    // Query the status of the node.
        Long nodeId = listNode(taskName);  // Query the ID of the node.
        RunCycleDagNodes(nodeId);   // Backfill data for the node.
    }
}
```

2.2.4. Synchronization scenario example

2.2.4.1. Synchronize incremental data

This topic describes how to synchronize incremental data from a Relational Database Service (RDS) database to MaxCompute. You can refer to this topic to synchronize incremental data in different scenarios.

Context

Data to be synchronized is divided into historical data and incremental data. Historical data, which is generally the log data, remains unchanged after it is written to the destination. Incremental data, such as changes of the staff status in the staff table, dynamically changes after it is written to the destination.

If the execution results of a node remain the same when you run the node multiple times, you can schedule the node to rerun it. If an error occurs in the node, you can clear dirty data. This principle is called idempotence. Each time when you write data, the data is written to a separate table or partition or overwrites the historical data in an existing table or partition.

In this topic, the scheduled date of a sync node is set to November 14, 2016, and historical data is synchronized to the ds=20161113 partition on the same day. In the incremental synchronization scenario, automatic scheduling is configured to synchronize incremental data to the ds=20161114 partition on the early morning of November 15. The optime field indicates the time when the data is modified. You can use this field to determine whether the data is incremental data.

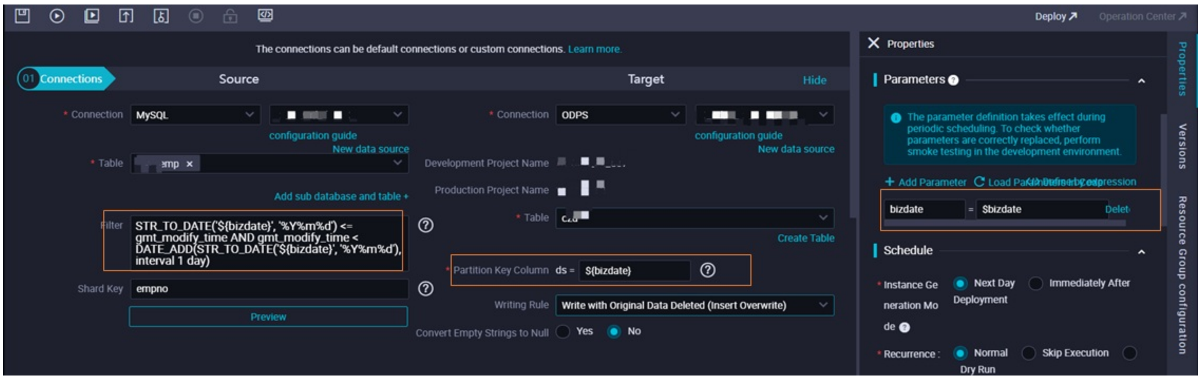
Usage notes

- Incremental synchronization is not supported for some data sources, such as HBase and OTSStream. You can refer to the topics that introduce the Reader plug-ins of the related data sources to check whether incremental synchronization is supported.
- You may need to set different parameters if you use different Reader plug-ins to synchronize incremental data. For more information, see Supported data source types, readers, and writers. The following table provides examples of the required parameters and supported syntax.

Reader plug-in	Parameter required for incremental synchronization	Supported syntax
MySQL	where Note If you configure a sync node by using the codeless UI, you must set the Filter parameter.	Use the syntax of the database. Note You can set scheduling parameters to read the data that is generated during a specified period of time every day.
MongoDB	query Note If you configure a sync node by using the codeless UI, you must set the Search Condition parameter.	Use the syntax of the database. Note You can set scheduling parameters to read the data that is generated during a specified period of time every day.
OSS	Object	Specify the object path. Note You can set scheduling parameters to read data from specified objects every day.
...	...	...

- The scheduling parameters are automatically set based on the data timestamp of the sync node. This way, incremental data generated each day is synchronized. For more information about scheduling parameters, see Overview of scheduling parameters.

  In the following example, the daily incremental data of a MySQL database is written to the corresponding partition of a MaxCompute table.



The connections can be default connections or custom connections. Learn more.

01 Connections

Source

- * Connection: MySQL [redacted]
- * Table: [redacted]emp ×
- Filter:

```
STR_TO_DATE('${bizdate}', '%Y%m%d') <=
gmt_modify_time AND gmt_modify_time <
DATE_ADD(STR_TO_DATE('${bizdate}', '%Y%m%d'),
interval 1 day)
```

- Shard Key: empno

Target

- * Connection: ODPS [redacted]
- Development Project Name: [redacted]
- Production Project Name: [redacted]
- * Table: [redacted]
- * Partition Key Column: ds = ${bizdate}
- Writing Rule: Write with Original Data Deleted (Insert Overwrite)
- Convert Empty Strings to Null: ○ Yes ● No

Properties

Parameters

The parameter definition takes effect during periodic scheduling. To check whether parameters are correctly replaced, perform smoke testing in the development environment.

bizdate = $bizdate

Schedule

- * Instance Generation Mode: ● Next Day ○ Immediately After Deployment
- * Recurrence : ● Normal ○ Skip Execution ○ Dry Run

Create a workflow

1. Log on to the DataWorks console.
2. In the left-side navigation pane, click **Workspaces**.
3. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
4. In the **Scheduled Workflow** pane, right-click **Business Flow** and select **Create Workflow**.
5. In the **Create Workflow** dialog box, set the **Workflow Name** and **Description** parameters.

   > **Note** The workflow name must be 1 to 128 characters in length.

6. Click **Create**.

Synchronize unchanged historical data in incremental mode

Historical data does not change after it is generated. Therefore, you can partition a table based on the pattern in which data is generated. Typically, you can partition a table by date, such as one partition per day.

1. Execute the following statements in the RDS database to prepare data:

   ```
   drop table if exists oplog;
   create table if not exists oplog(
   optime DATETIME,
   uname varchar(50),
   action varchar(50),
   status varchar(10)
    );
   Insert into oplog values(str_to_date('2016-11-11','%Y-%m-%d'),'LiLei','SELECT','SUCCESS');
   Insert into oplog values(str_to_date('2016-11-12','%Y-%m-%d'),'HanMM','DESC','SUCCESS');
   ```

   The preceding two data entries are used as historical data. You must synchronize all the historical data to the ds=20161113 partition first.

2. In the **Scheduled Workflow** pane, expand the created workflow, right-click **Table** under MaxCompute, and then select **Create Table**.
3. In the **Create Table** dialog box, set the **Table Name** parameter to ods_oplog and click **Create**.
4. On the configuration tab of the ods_oplog table on the right side, click **DDL Statement**. In the DDL Statement dialog box, enter the following statement to create a MaxCompute table:

   ```
   -- Create a MaxCompute table and partition the table by day.
   create table if not exists ods_oplog(
    optime datetime,
    uname string,
    action string,
    status string
   ) partitioned by (ds string);
   ```

5. Configure a sync node to synchronize historical data. For more information, see Create a synchronization node.

   After you test the sync node, click **Properties** in the right-side navigation pane. In the Properties panel, select **Skip Execution** and commit or deploy the node again to prevent the node from being automatically scheduled.

6. Execute the following statements to insert data into the source RDS table as incremental data:

   ```
   insert into oplog values(CURRENT_DATE,'Jim','Update','SUCCESS');
   insert into oplog values(CURRENT_DATE,'Kate','Delete','Failed');
   insert into oplog values(CURRENT_DATE,'Lily','Drop','Failed');
   ```

7. Configure a sync node to synchronize incremental data.

   In the **Source** section, enter `date_format(optime,'%Y%m%d')=${bdp.system.bizdate}` in the Filter field. In the **Target** section, enter `${bdp.system.bizdate}` in the Partition Key Column field.

   > **Note** By setting a data filter, you can query data that is inserted into the source table on November 14 and synchronize the data to the incremental data partition of the destination table on the early morning of November 15.

8. View the incremental synchronization result.

   Click the **Properties** tab in the right-side navigation pane. In the Properties panel, set the Scheduling Cycle parameter to Day. After you commit or deploy the incremental sync node, the node is automatically scheduled to run from the next day. After the node is run, you can view data in the destination MaxCompute table.

Synchronize dynamically updated data in incremental mode

Because of the time-variant characteristic of data warehouses, we recommend that you daily synchronize all data in tables that are subject to changes, such as staff and order tables. In other words, full data is stored on a daily basis. This way, you can retrieve historical and current data.

In actual scenarios, you may synchronize only incremental data every day under special circumstances. MaxCompute does not support editing data by using the UPDATE statement. Therefore, you can only use other methods to synchronize data. The following section describes how to synchronize data in full mode and in incremental mode.

1. Execute the following statements to prepare data:

```
drop table if exists user ;
create table if not exists user(
    uid int,
    uname varchar(50),
    deptno int,
    gender VARCHAR(1),
    optime DATETIME
    );
-- Insert historical data.
insert into user values (1,'LiLei',100,'M',str_to_date('2016-11-13','%Y-%m-%d'));
insert into user values (2,'HanMM',null,'F',str_to_date('2016-11-13','%Y-%m-%d'));
insert into user values (3,'Jim',102,'M',str_to_date('2016-11-12','%Y-%m-%d'));
insert into user values (4,'Kate',103,'F',str_to_date('2016-11-12','%Y-%m-%d'));
insert into user values (5,'Lily',104,'F',str_to_date('2016-11-11','%Y-%m-%d'));
-- Insert incremental data.
update user set deptno=101,optime=CURRENT_TIME  where uid = 2; -- Change null to non-null.
update user set deptno=104,optime=CURRENT_TIME  where uid = 3; -- Change non-null to non-null.
update user set deptno=null,optime=CURRENT_TIME  where uid = 4; -- Change non-null to null.
delete from user where uid = 5;
insert into user(uid,uname,deptno,gender,optime) values (6,'Lucy',105,'F',CURRENT_TIME);
```

2. Synchronize data.

   - Daily synchronize all data.

     a. Execute the following statement to create a MaxCompute table. For more information, see Create a MaxCompute table.

```
-- Synchronize all data.
create table ods_user_full(
    uid bigint,
    uname string,
    deptno bigint,
    gender string,
    optime DATETIME
) partitioned by (ds string);
```

     b. Configure a sync node to synchronize all data.

        Note Set the Scheduling Cycle parameter to Day because daily full data synchronization is required.

     c. Run the sync node and view data in the destination MaxCompute table after the synchronization is complete.

        When full data synchronization is performed on a daily basis, no incremental synchronization is performed. You can view the data results in the table after the node is automatically scheduled to run on the next day.

   - Daily synchronize incremental data.

We recommend that you do not use this sync mode except in scenarios in which the DELETE statement is not supported and you fail to execute relevant SQL statements to view deleted data. Generally, your enterprise code is logically deleted, in which the UPDATE statement is applied instead of the DELETE statement. In scenarios in which this method is inapplicable, this sync mode may cause data inconsistency if a special condition is encountered. Another drawback is that you must merge new and historical data after the synchronization.

Prepare data

Create two tables, one for writing all the latest data and the other for writing incremental data.

```
-- Create a result table.
create table dw_user_inc(
    uid bigint,
    uname string,
    deptno bigint,
    gender string,
    optime DATETIME
);
```

```
-- Create an incremental data table.
create table ods_user_inc(
    uid bigint,
    uname string,
    deptno bigint,
    gender string,
    optime DATETIME
)
```

a. Configure a sync node to write all data to the result table.

> Note You need only to run the node once. After you run the node, click the **Properties** tab in the right-side navigation pane. In the Properties panel, select **Skip Execution** for the Recurrence parameter.

b. Configure a sync node to write incremental data to the incremental data table. To filter the data, set the **where** parameter to `date_format(optime,'%Y%m%d')=${bdp.system.bizdate}` .

c. Execute the following statement to merge data:

```
insert overwrite table dw_user_inc
select
-- All the SELECT clauses are listed. If the incremental data table contains data, data in the result table changes. In this case, use data in the incremental data table.
case when b.uid is not null then b.uid else a.uid end as uid,
case when b.uid is not null then b.uname else a.uname end as uname,
case when b.uid is not null then b.deptno else a.deptno end as deptno,
case when b.uid is not null then b.gender else a.gender end as gender,
case when b.uid is not null then b.optime else a.optime end as optime
from
dw_user_inc a
full outer join ods_user_inc b
on a.uid  = b.uid ;
```

View the merge result. It is found that the deleted data entry is not synchronized.

Daily incremental synchronization is advantageous because it synchronizes only a small amount of incremental data. However, it may cause data inconsistency, which requires an extra computing workload to merge data.

If not necessary, daily synchronize dynamically updated data in full mode. In addition, you can set a lifecycle for the historical data so that it can be automatically deleted after being retained for a specific period.

2.2.4.2. Synchronize data from tables in multiple databases to a specific table

Data Integration allows you to synchronize data from tables in multiple databases to a specific table at a time. Specifically, you can create a batch sync node, specify tables in multiple databases as the source tables, and then specify the destination table. After that, you can run the batch sync node to synchronize the data from the source tables to the destination table.

Context

When you configure a batch sync node for synchronizing data from tables in multiple databases to a specific table, you must specify source tables. Make sure that all the source tables have the same schema.

You can specify source tables in various types of databases such as MySQL, SQL Server, Oracle, PostgreSQL, PolarDB, and AnalyticDB databases. You can use the codeless user interface (UI) to configure a batch sync node for synchronizing data from tables in multiple MySQL databases to a specific table.

Procedure

1. Go to the **DataStudio** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **DataStudio** in the Actions column.
2. Create a batch sync node.
   i. Move the pointer over the **+Create** icon and choose **Data Integration > Batch Synchronization**.

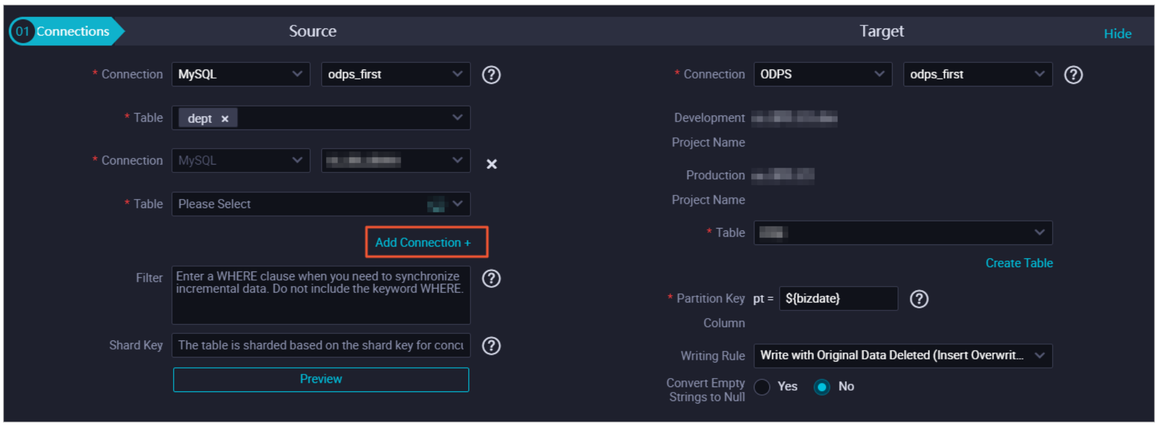
      Alternatively, you can click the desired workflow in the Business Flow pane, right-click **Data Integration**, and then choose **Create > Batch Synchronization**.
   ii. In the **Create Node** dialog box, set the **Node Name** and **Location** parameters.

      > Note The node name must be 1 to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).
   iii. Click **Commit**.
3. Configure the batch sync node.

   You can configure the batch sync node on the codeless UI or in the code editor.

   - If you configure the batch sync node on the codeless UI, specify the source databases and tables in the **Connections** section. For more information, see Configure a synchronization node by using the codeless UI.

     > Note You can use the codeless UI to configure a batch sync node for synchronizing data from tables only in multiple MySQL databases. For more information about how to use the code editor to configure batch sync nodes, see Create a synchronization node by using the code editor.

     01 Connections

     Source

     * Connection: MySQL, odps_first
     * Table: dept ×
     * Connection: MySQL, [redacted] ×
     * Table: Please Select

     Add Connection +

     Filter: Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.

     Shard Key: The table is sharded based on the shard key for concu

     Preview

     Target

     Hide

     * Connection: ODPS, odps_first
     Development Project Name: [redacted]
     Production Project Name: [redacted]
     * Table: [redacted]

     Create Table

     * Partition Key Column: pt = ${bizdate}
     Writing Rule: Write with Original Data Deleted (Insert Overwrit...
     Convert Empty Strings to Null: ○ Yes ● No

     If you want to add data sources, click **New data source**. For more information, see Add a MySQL data source.
   - If you configure the batch sync node in the code editor, write the code for synchronizing data from the source tables in multiple databases to a specific table. The following code provides an example: For more information, see Create a synchronization node by using the code editor.

Notice Delete the comments from the following code before you run the code:

```
{
    "type": "job",
    "version": "1.0",
    "configuration": {
        "reader": {
            "plugin": "mysql",
            "parameter": {
                "connection": [
                    {
                        "table": [ // The source tables in the first database.
                            "tbl1",
                            "tbl2",
                            "tbl3"
                        ],
                        "datasource": "datasourceName1" // The name of the first database.
                    },
                    {
                        "table": [ // The source tables in the second database.
                            "tbl4",
                            "tbl5",
                            "tbl6"
                        ],
                        "datasource": "datasourceName2" // The name of the second database.
                    }
                ],
                "singleOrMulti": "multi",
                "splitPk": "db_id",
                "column": [
                    "id", "name", "age"
                ],
                "where": "1 < id and id < 100"
            }
        },
        "writer": {
        }
    }
}
```

4. Commit the node.

   Notice You must set the **Rerun** and **Parent Nodes** parameters before you can commit the node.

   i. Click the icon in the toolbar.

   ii. In the **Commit Node** dialog box, enter your comments in the **Change description** field.

   iii. Click **OK**.

   In a workspace in standard mode, you must click **Deploy** in the upper-right corner after you commit the do-while node.

5. Test the node. For more information, see View auto triggered nodes.

2.3. Database migration and migration of multiple tables to the cloud

The database migration feature allows you to upload all tables in the source data store to the destination data store in an efficient and cost-effective manner. This saves the time spent on creating multiple nodes one by one to migrate your initial data to the cloud.

Supported data stores

DataWorks allows you to migrate data from a variety of data stores to MaxCompute by using the database migration feature. The following table describes the data stores that the database migration feature supports and the references.

Destination	Source	References
MaxCompute	MySQL PostgreSQL SQL Server Oracle PolarDB AnalyticDB for MySQL V2.0 AnalyticDB for MySQL V3.0 AnalyticDB for PostgreSQL HybridDB for MySQL DRDS DM Hive DB2	Synchronize full data to MaxCompute on a regular basis Synchronize incremental data to MaxCompute on a regular basis Synchronize full data to MaxCompute at a time Synchronize incremental data to MaxCompute at a time Synchronize full data to MaxCompute at a time and then synchronize incremental data on a regular basis

2.4. Node optimization

2.4.1. Optimize performance

This topic describes the factors that affect the speed of data synchronization, and how to adjust the concurrency for sync nodes to maximize the synchronization speed. This topic also describes bandwidth throttling settings and scenarios of slow data synchronization.

Data Integration of DataWorks is a one-stop platform that supports real-time and batch data synchronization between data stores in all regions and all network environments. You can synchronize data between various types of cloud storage and local storage each day.

DataWorks provides excellent data transmission performance and supports data synchronization between more than 400 pairs of heterogeneous data stores. These features allow you to focus on the key issues on constructing big data solutions.

Factors that affect the speed of data synchronization

The following factors affect the speed of data synchronization:

- Source data store
  - Database performance: the performance of the CPU, memory, solid-state drive (SSD), network, and hard disk.
  - Concurrency: A high concurrency results in a heavy database workload.
  - Network: the bandwidth (throughput) and speed of the network. Generally, a database that has better performance can support more concurrent nodes and a greater concurrency value can be set for sync nodes.
- Configuration of sync nodes
  - Synchronization speed: whether an upper limit is set for the synchronization speed.
  - Concurrency: the maximum number of concurrent threads to read data from the source data store and write data to the destination data store.
  - Nodes that are waiting for resources.
  - Bandwidth throttling: The bandwidth of a single thread is 1,048,576 bit/s. Timeout occurs when the business is sensitive to the network speed. We recommend that you set a small bandwidth limit.
  - Whether an index is created for query statements.
- Destination data store
  - Performance: the performance of the CPU, memory, SSD, network, and hard disk.
  - Load: Excessive load in the destination database affects the write efficiency within the sync nodes.
  - Network: the bandwidth (throughput) and speed of the network.

You must monitor and optimize the performance, load, and network of the source and destination databases. The following sections describe the optimal configuration of a sync node in Data Integration.

Concurrency

You can configure the concurrency for a sync node on the codeless user interface (UI). The following example shows how to configure the concurrency in the code editor:

```
"setting": {
      "speed": {
        "concurrent": 10
      }
    }   }
```

Bandwidth throttling

By default, bandwidth throttling is disabled. In a sync node, data is synchronized at the maximum transmission rate given the concurrency that is configured for the node. Excessively fast synchronization may overstress the database and thus affect the production. Therefore, Data Integration allows you to limit the synchronization speed and optimize the configuration as required. If bandwidth throttling is enabled, we recommend that you limit the maximum transmission rate to 30 Mbit/s. The following example shows how to configure an upper limit for the synchronization speed in the code editor, in which the transmission bandwidth is 1 Mbit/s:

```
"setting": {
      "speed": {
         "throttle": true // Specifies that bandwidth throttling is enabled.
        "mbps": 1, // The synchronization speed.
      }
    }
```

- The valid values of the throttle parameter are *true* and *false*.
    - If you set the throttle parameter to *true*, bandwidth throttling is enabled. In this case, you must set the mbps parameter. If you do not set the mbps parameter, an error is returned when the sync node is run or data is synchronized at an abnormal speed.
    - If you set the throttle parameter to *false*, bandwidth throttling is disabled, and you do not need to set the mbps parameter.
- The bandwidth value is a Data Integration metric and does not represent the actual network interface card (NIC) traffic. Generally, the NIC traffic is two to three times of the channel traffic. The actual NIC traffic depends on the serialization of the data storage system.
- A semi-structured file does not have shard keys. If multiple files exist, you can set the maximum transmission rate of a node to increase the synchronization speed. However, the maximum transmission rate is limited by the number of files.

  Assume that the maximum transmission rate can be set to n Mbit/s for n files.

    - If you set the maximum transmission rate to (n + 1) Mbit/s, the files are still synchronized at a speed of n Mbit/s.
    - If you set the maximum transmission rate to (n - 1) Mbit/s, the files are synchronized at a speed of (n - 1) Mbit/s.
- A table in a relational database can be split based on the maximum transmission rate only after you set the maximum transmission rate and shard key. In general, relational databases support only numeric-type shard keys. However, Oracle databases support numeric- and string-type shard keys.

Scenarios of slow data synchronization

- Scenario 1: Sync nodes to be run on shared resources for scheduling remain waiting for resources.

- Sample scenario

  When you test a node in DataWorks, the node remains waiting for resources and an internal system error occurs.

  For example, you use the default resource group to run a sync node to synchronize data from ApsaraDB Relational Database Service (RDS) to MaxCompute. The node has waited about 800 seconds before it is run. However, the log shows that the node runs for only 18 seconds. When you run other sync nodes, they also remain in the waiting state.

  The following log is displayed:

  ```
  2017-01-03 07:16:54 : State: 2(WAIT) | Total: 0R 0B | Speed: 0R/s 0B/s | Error: 0R 0B | Stage: 0.0%
  ```

- Solution

  The shared resources for scheduling that you use are not exclusively used by a single user. Many nodes, not only two or three nodes of a single user, are run on the shared resources for scheduling. If resources are insufficient after you start to run a node, the node must wait for resources. In this case, the node is delayed for 800 seconds, and it takes only 10 seconds for the node to be run.

  To improve the synchronization speed and reduce the waiting time, we recommend that you run sync nodes during off-peak hours. Generally, most sync nodes are run between 00:00 and 03:00. You can avoid this time period to prevent your nodes from waiting for resources.

- Scenario 2: Accelerate nodes that synchronize data from multiple source tables to the same destination table.
  - Sample scenario

    Sync nodes are configured to run in sequence to synchronize data from tables of multiple data stores to the same destination table. However, the synchronization takes a long time.

  - Solution

    To start multiple concurrent nodes that write data to the same destination database, take note of the following points:

    - Make sure that the destination database can support the running of all the concurrent nodes.
    - You can configure a sync node that synchronizes data from multiple source tables to the same destination table. Alternatively, you can configure multiple nodes to concurrently run in the same workflow.
    - If nodes need to wait for resources when they are run, you can configure them to run during off-peak hours. This ensures that the nodes have a higher execution priority.

- Scenario 3: A full table scan slows down the data synchronization because no index is added in the WHERE clause.
  - Sample scenario

    The following SQL statement is executed:

    ```
    select bid,inviter,uid,createTime from `relatives` where createTime>='2016-10-2300:00:00'and reateTime<'2016-10-24 00:00:00';
    ```

    The sync node started to run at 11:01:24.875 on October 25, 2016 and started to return results at 11:11:05.489 on October 25, 2016. The synchronization program is waiting for the database to return SQL query results. However, it takes a long time before MaxCompute can respond.

  - Cause

    When the WHERE clause is used for a query, the createTime column is not indexed, resulting in a full table scan.

  - Solution

    If you use the WHERE clause, we recommend that you use an indexed column or add an index to the column that you want to scan. This can improve performance.

2.4.2. Optimize a sync node

When a sync node is scheduled, the instance may take longer than expected to run. This topic describes how to optimize a sync node when the instance runs slowly or the node start time greatly differs from the scheduling time.

Prerequisites

The operational logs and attribute information of a sync node are obtained before you optimize the node.

DataWorks provides level-1 scheduling resources and level-2 running resources for sync nodes.

- Level-1 scheduling resources: Go to **Operation Center** and choose **Cycle Task Maintenance > Cycle Task** in the left-side navigation pane. On the page that appears, right-click the sync node in the directed acyclic graph (DAG) on the right and select **View Node Details**. On the **Node Information** page that appears, you can view the attribute information and level-1 scheduling resources of the node.
- Level-2 running resources: Go to the **Data Integration** module and click **Custom Resource Group** in the left-side navigation pane. On the page that appears, you can view and add level-2 running resources.

Context

Generally, a sync node may be considered slow in the following scenarios:

- The node start time greatly differs from the scheduling time.
- The sync node remains in the WAIT state for a long time.
- The sync node runs at a low speed.

Scenario 1 where the start time of a sync node greatly differs from the scheduling time

In this scenario, you must first obtain the operational logs and attribute information of the node and compare the operational logs with the attribute information. The comparison result shows that the node start time in the operational logs differs from the scheduling time in the attribute information. Most of the time is consumed while waiting for scheduling.

Problem example

1. Go to **Operation Center** and choose **Cycle Task Maintenance > Cycle Task** in the left-side navigation pane. On the page that appears, right-click the sync node in the DAG on the right and select **View Node Details**. On the **Node Information** page that appears, check the scheduling time of the node, which is 00:00. However, the node actually starts at 00:29. It is inferred that most of the time is consumed while waiting for scheduling.
2. Choose **Cycle Task Maintenance > Cycle Instance** in the left-side navigation pane. On the page that appears, right-click the node instance and select **View Runtime Log**. The operational logs show that the node starts at 00:29 and ends at 00:30. That is, it takes only 1 minute to complete the task. This indicates that the node is able to run normally.

Troubleshooting

1. Check whether multiple nodes are scheduled at the same time in your workspace. The default resource group contains a limited number of level-1 scheduling resources. If multiple nodes are scheduled at the same time, other nodes must queue up and wait for scheduling.
2. The peak hours for business scheduling range from 00:00 to 02:00. We recommend that you run your business during off-peak hours.

Scenario 2 where a sync node runs at a steady speed of 0

The operational logs of the node show that the node runs at a steady speed of 0. This is usually because the source database has a high CPU load or network traffic usage, and therefore the SQL statement is executed slowly. Or, truncate operations are performed before the SQL statement is executed, and therefore the SQL statement is processed for a long time.

Problem example

1. The operational logs of the node show that the node keeps running from 18:00 to 21:13 at a steady speed of 0.
2. The operational logs also record a truncate operation that lasts from 18:00 to 21:13.

Troubleshooting

It can be inferred that the sync node is slow due to the truncate operation. You need to check the cause of the slow truncate operation in the source database.

Scenario 3 where a sync node runs at a low speed

The operational logs of the node show that the node runs at a low speed slightly greater than 0.

Problem example

1. The operational logs of the node show that the node runs at a low speed of about 1.93 Kbit/s.
2. The value of the WaitReaderTime field is greater than that of the WaitWriterTime field, indicating that more time is consumed for waiting to read data.

Troubleshooting

If the node runs at a low speed, check whether the value of WaitReaderTime or WaitWriterTime is large. If reading or writing consumes more time, check the load of the corresponding source or destination database.

2.5. Resources for batch synchronization nodes

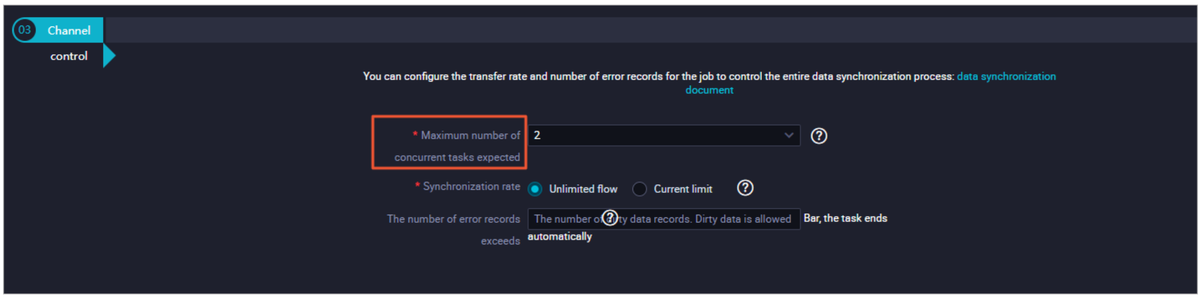
This topic describes the concurrency configurations and synchronization speed of batch synchronization nodes. This topic also describes the relationships between concurrency and resource usage.

Methods for configuring the concurrency

To adjust the resource usage and synchronization speed of a batch synchronization node, you can configure the concurrency for the batch synchronization node by using one of the following methods:

- Configure the concurrency for a batch synchronization node by using the codeless user interface (UI). For more information, see Configure a synchronization node by using the codeless UI.

  On the configuration tab of the batch synchronization node, configure the **Expected Maximum Concurrency** parameter in the **Channel** section.



You can configure the transfer rate and number of error records for the job to control the entire data synchronization process: data synchronization document

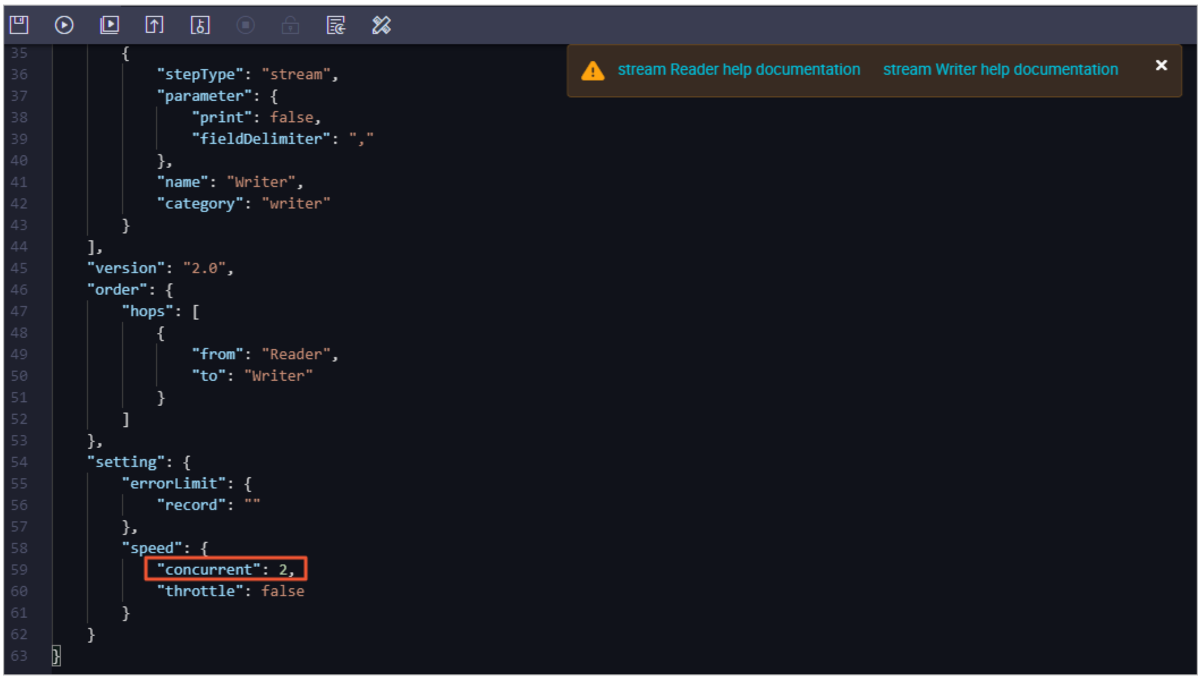
* Maximum number of concurrent tasks expected: 2

* Synchronization rate: ● Unlimited flow ○ Current limit

The number of error records exceeds: The number of dirty data records. Dirty data is allowed Bar, the task ends automatically

- Configure the concurrency for a batch synchronization node by using the code editor. For more information, see Create a synchronization node by using the code editor.

  In the code editor, configure *$.setting.speed.concurrent* in the JSON structure.



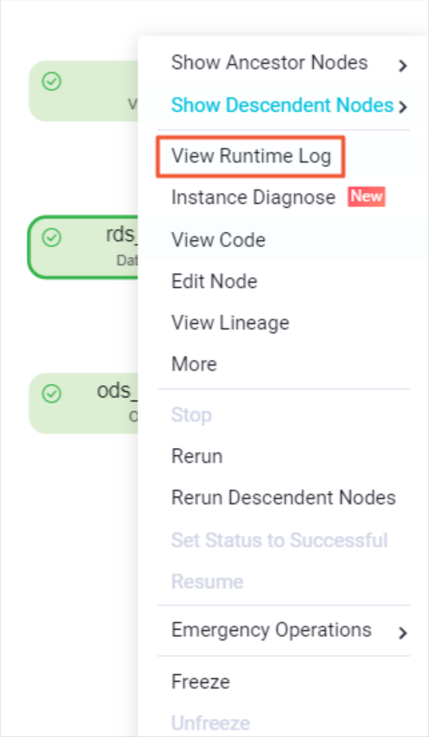
stream Reader help documentation
stream Writer help documentation

```
        {
            "stepType": "stream",
            "parameter": {
                "print": false,
                "fieldDelimiter": ","
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": ""
        },
        "speed": {
            "concurrent": 2,
            "throttle": false
        }
    }
}
```

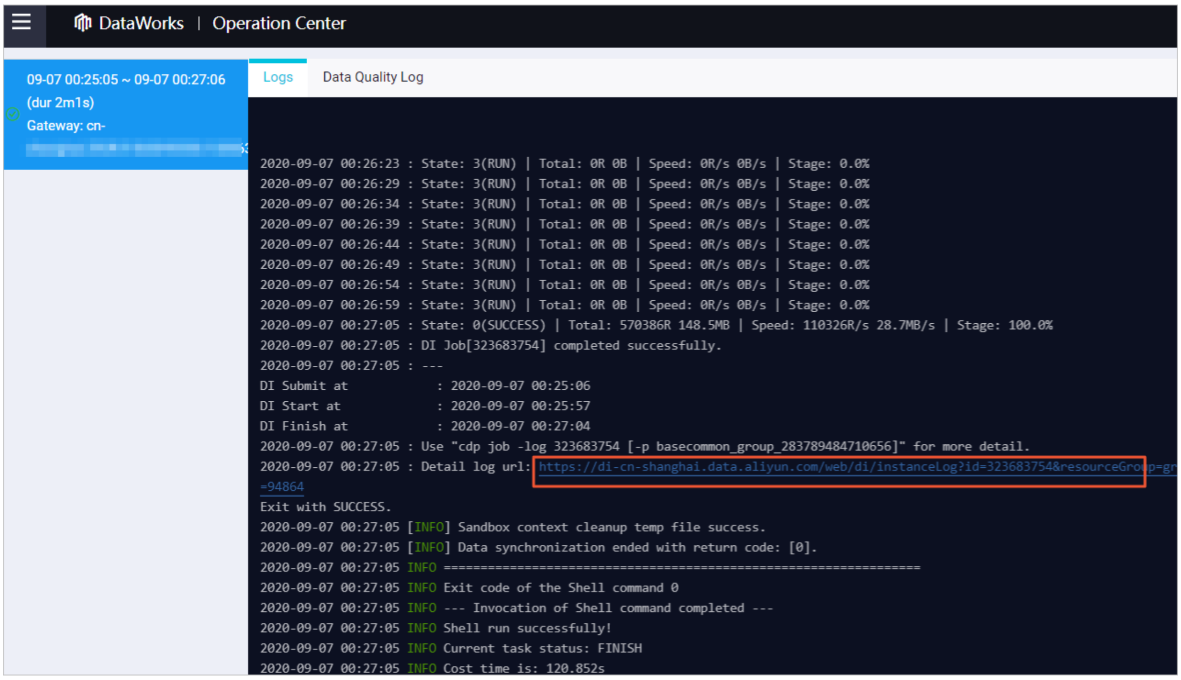
In consideration of system performance and limits on data sources, the actual concurrency of the batch synchronization node may be less than the concurrency specified by using the codeless UI or code editor.

To check the actual concurrency of the batch synchronization node, perform the following steps:

1. Log on to the DataWorks console.
2. In the left-side navigation pane, click **Workspaces**.
3. In the top navigation bar, select the region where your workspace resides. Find your workspace and choose More > **Operation Center** in the Actions column.
4. In the left-side navigation pane, choose **Cycle Task Maintenance > Cycle Instance**.
5. Click a desired instance of the batch synchronization node. The directed acyclic graph (DAG) of the node instance appears on the right. In the DAG, right-click the node instance and select **View Runtime Log**.



6. On the page that appears, click the link next to **Detail log url**.

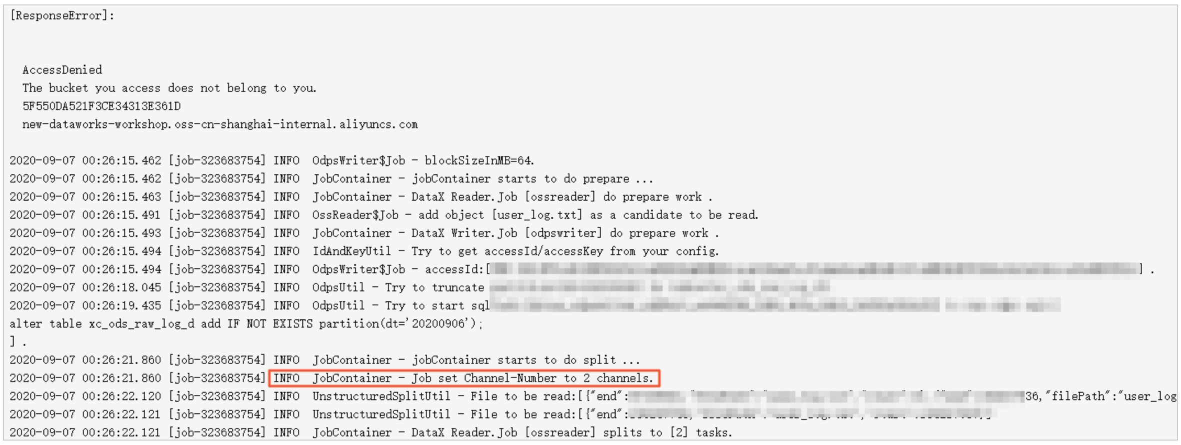


09-07 00:25:05 ~ 09-07 00:27:06
(dur 2m1s)
Gateway: cn-
[redacted]

```
2020-09-07 00:26:23 : State: 3(RUN) | Total: 0R 0B | Speed: 0R/s 0B/s | Stage: 0.0%
2020-09-07 00:26:29 : State: 3(RUN) | Total: 0R 0B | Speed: 0R/s 0B/s | Stage: 0.0%
2020-09-07 00:26:34 : State: 3(RUN) | Total: 0R 0B | Speed: 0R/s 0B/s | Stage: 0.0%
2020-09-07 00:26:39 : State: 3(RUN) | Total: 0R 0B | Speed: 0R/s 0B/s | Stage: 0.0%
2020-09-07 00:26:44 : State: 3(RUN) | Total: 0R 0B | Speed: 0R/s 0B/s | Stage: 0.0%
2020-09-07 00:26:49 : State: 3(RUN) | Total: 0R 0B | Speed: 0R/s 0B/s | Stage: 0.0%
2020-09-07 00:26:54 : State: 3(RUN) | Total: 0R 0B | Speed: 0R/s 0B/s | Stage: 0.0%
2020-09-07 00:26:59 : State: 3(RUN) | Total: 0R 0B | Speed: 0R/s 0B/s | Stage: 0.0%
2020-09-07 00:27:05 : State: 0(SUCCESS) | Total: 570386R 148.5MB | Speed: 110326R/s 28.7MB/s | Stage: 100.0%
2020-09-07 00:27:05 : DI Job[323683754] completed successfully.
2020-09-07 00:27:05 : ---
DI Submit at            : 2020-09-07 00:25:06
DI Start at             : 2020-09-07 00:25:57
DI Finish at            : 2020-09-07 00:27:04
2020-09-07 00:27:05 : Use "cdp job -log 323683754 [-p basecommon_group_283789484710656]" for more detail.
2020-09-07 00:27:05 : Detail log url: https://di-cn-shanghai.data.aliyun.com/web/di/instanceLog?id=323683754&resourceGroup=gr
=94864
Exit with SUCCESS.
2020-09-07 00:27:05 [INFO] Sandbox context cleanup temp file success.
2020-09-07 00:27:05 [INFO] Data synchronization ended with return code: [0].
2020-09-07 00:27:05 INFO ================================================================
2020-09-07 00:27:05 INFO Exit code of the Shell command 0
2020-09-07 00:27:05 INFO --- Invocation of Shell command completed ---
2020-09-07 00:27:05 INFO Shell run successfully!
2020-09-07 00:27:05 INFO Current task status: FINISH
2020-09-07 00:27:05 INFO Cost time is: 120.852s
```

7. On the log details page of the batch synchronization node, find the entry in the `JobContainer - Job set Channel-Number to X channels` format. In this example, the value of X is 2, which indicates that two concurrent threads are used by

the batch synchronization node.



```
[ResponseError]:


  AccessDenied
  The bucket you access does not belong to you.
  5F550DA521F3CE34313E361D
  new-dataworks-workshop.oss-cn-shanghai-internal.aliyuncs.com

2020-09-07 00:26:15.462 [job-323683754] INFO  OdpsWriter$Job - blockSizeInMB=64.
2020-09-07 00:26:15.462 [job-323683754] INFO  JobContainer - jobContainer starts to do prepare ...
2020-09-07 00:26:15.463 [job-323683754] INFO  JobContainer - DataX Reader.Job [ossreader] do prepare work .
2020-09-07 00:26:15.491 [job-323683754] INFO  OssReader$Job - add object [user_log.txt] as a candidate to be read.
2020-09-07 00:26:15.493 [job-323683754] INFO  JobContainer - DataX Writer.Job [odpswriter] do prepare work .
2020-09-07 00:26:15.494 [job-323683754] INFO  IdAndKeyUtil - Try to get accessId/accessKey from your config.
2020-09-07 00:26:15.494 [job-323683754] INFO  OdpsWriter$Job - accessId:[[redacted]] .
2020-09-07 00:26:18.045 [job-323683754] INFO  OdpsUtil - Try to truncate [redacted]
2020-09-07 00:26:19.435 [job-323683754] INFO  OdpsUtil - Try to start sql[redacted]
alter table xc_ods_raw_log_d add IF NOT EXISTS partition(dt='20200906');
] .
2020-09-07 00:26:21.860 [job-323683754] INFO  JobContainer - jobContainer starts to do split ...
2020-09-07 00:26:21.860 [job-323683754] INFO  JobContainer - Job set Channel-Number to 2 channels.
2020-09-07 00:26:22.120 [job-323683754] INFO  UnstructuredSplitUtil - File to be read:[{"end":[redacted]36,"filePath":"user_log
2020-09-07 00:26:22.121 [job-323683754] INFO  UnstructuredSplitUtil - File to be read:[{"end":[redacted]
2020-09-07 00:26:22.121 [job-323683754] INFO  JobContainer - DataX Reader.Job [ossreader] splits to [2] tasks.
```

Relationships between the concurrency and resource usage

This section describes the relationships between the concurrency and CPU utilization and between the concurrency and memory usage for exclusive resource groups.

- Relationship between the concurrency and CPU utilization

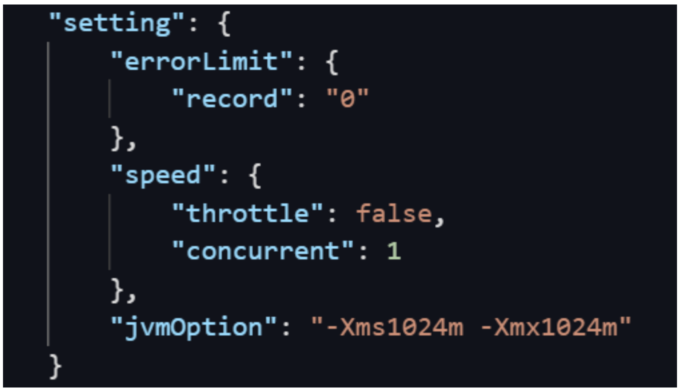
  For exclusive resource groups, the ratio of the concurrency to CPU utilization is 1:0.5. For example, if the exclusive resource group that you purchase uses an Elastic Compute Service (ECS) instance with the specifications of 4 vCPUs and 8 GiB of memory, the concurrency of the exclusive resource group is 8. You can run a maximum of eight batch synchronization nodes with a concurrency of 1 or four batch synchronization nodes with a concurrency of 2 at a time.

  If the node that you want to run on the exclusive resource group requires more threads than the threads available in the exclusive resource group, the node needs to wait until one or more nodes stop running and sufficient threads are available for the node.

  > **Note** If the node that you want to run on the exclusive resource group requires more threads than the maximum number of threads that can be provided by the exclusive resource group, the node enters the state of waiting for resources. For example, if you want to run a node that requires 10 concurrent threads on the exclusive resource group that uses an ECS instance with the specifications of 4 vCPUs and 8 GiB of memory, the node will permanently wait for resources. The exclusive resource group allocates resources to nodes based on the sequence in which the nodes are committed. Therefore, DataWorks cannot run nodes that are committed later than this node.

- Relationship between the concurrency and memory usage

  In an exclusive resource group, the minimum memory size that can be allocated to a synchronization node is calculated by using the following formula: 768 MB + (Concurrency - 1) × 256 MB. The maximum memory size that can be allocated to a synchronization node is 8,029 MB. However, if you specify the memory size required by a synchronization node when you configure the synchronization node, the specified memory size overrides the default settings of the exclusive resource group. When you configure a synchronization node by using the code editor, you can specify the memory size by configuring *$.setting.jvmOption* in the JSON structure.



```
"setting": {
    "errorLimit": {
        "record": "0"
    },
    "speed": {
        "throttle": false,
        "concurrent": 1
    },
    "jvmOption": "-Xms1024m -Xmx1024m"
}
```

To ensure smooth running of all the nodes that are run on an exclusive resource group, the total memory size used by all running nodes must be at least 1 GB less than the total memory size of all ECS instances deployed in the exclusive resource group. If this condition is not met, the Linux out of memory (OOM) killer forcibly stops the nodes that are running.

> **Note** If the required memory size is not modified in the code editor, you need to only consider the limits on the concurrency when you commit synchronization nodes.

Synchronization speed

The read and write speeds vary based on the involved data sources. This section describes the average speed for a thread to read data from or write data to each type of data source.

- Average speed for a thread to write data to each type of data source

Writer	Average write speed (KB per second)
AnalyticDB for PostgreSQL	147.8
AnalyticDB for MySQL	181.3
ClickHouse	5259.3
DataHub	45.8
PolarDB-X (DRDS)	93.1
Elasticsearch	74.0
FTP	565.6
GDB	17.1
HBase	2395.0
HBase20xsql	37.8
HDFS	1301.3
Hive	1960.4
HybridDB for MySQL	323.0
HybridDB for PostgreSQL	116.0
Kafka	0.9
LogHub	788.5
MongoDB	51.6
MySQL	54.9
MaxCompute	660.6
Oracle	66.7
OSS	3718.4
Tablestore	138.5
PolarDB	45.6
PostgreSQL	168.4
Redis	7846.7

Writer	Average write speed (KB per second)
SQL Server	8.3
Stream	116.1
TSDB	2.3
Vertica	272.0

- Average speed for a thread to read data from each type of data source

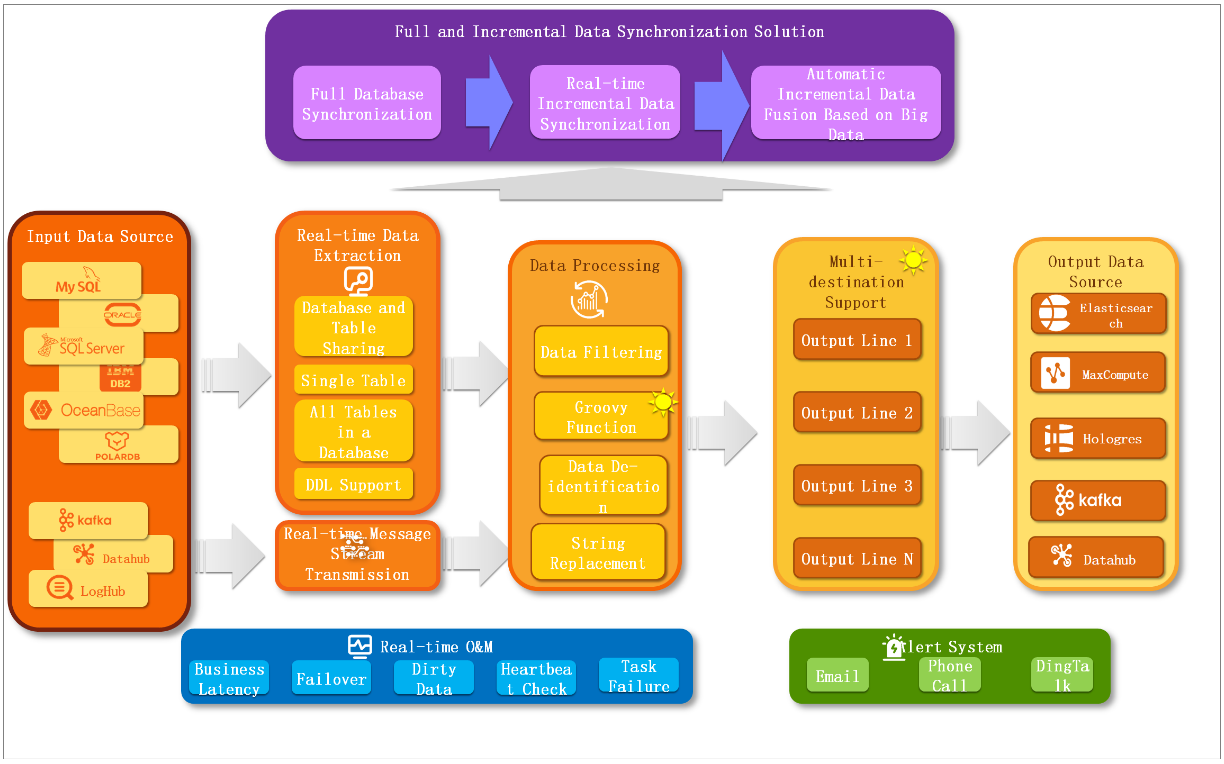
Reader	Average read speed (KB per second)
AnalyticDB for PostgreSQL	220.3
AnalyticDB for MySQL	248.6
PolarDB-X (DRDS)	146.4
Elasticsearch	215.8
FTP	279.4
HBase	1605.6
HBase20xsql	465.3
HDFS	2202.9
Hologres	741.0
HybridDB for MySQL	111.3
HybridDB for PostgreSQL	496.9
Kafka	3117.2
LogHub	1014.1
MongoDB	361.3
MySQL	459.5
MaxCompute	207.2
Oracle	133.5
OSS	665.3
Tablestore	229.3
OTSStream	661.7
PolarDB	238.2
PostgreSQL	165.6
RDBMS	845.6
SQL Server	143.7
Stream	85.0
Vertica	454.3

3.Real-time data synchronization

3.1. Overview of real-time synchronization nodes

DataWorks provides the real-time data synchronization feature. You can use this feature to synchronize data changes of a table or all tables in a source to a destination in real time. This way, data in the destination is consistent with data in the source in real time.

Architecture



ⓘ **Note** The Groovy plug-in and the feature to synchronize data to multiple destinations are in development and will be supported soon.

The real-time data synchronization feature has the following benefits:

- Diverse data sources

  Multiple types of data sources are supported. You can synchronize data between different types of data sources.

- Synchronization solutions

  You can configure a synchronization solution to synchronize the full data and then incremental data from a common source.

- Diverse synchronization methods

  You can synchronize data from table shards, a single table in a source, or multiple tables in a source, and configure different processing rules for messages about different DDL operations.

- Data processing

  You can perform **data filtering**, **string replacement** and **data masking** on the data from a source based on your business requirements and synchronize the processed data to a destination.

- Monitoring and alerting

The system can send you alert notifications about service latency, failover, dirty data, heartbeat, and failure by email, text message, or DingTalk message. This way, you can identify and handle alerts at the earliest opportunity.

> Note
>
> An alert notification can be sent by text message only in the Singapore (Singapore), Malaysia (Kuala Lumpur), and Germany (Frankfurt) regions. If you want to use this notification method in other regions, submit a ticket to contact Alibaba Cloud DataWorks technical support.

- Graphical development

  You can perform drag-and-drop operations instead of writing code to develop real-time synchronization nodes. It is easy to use for beginners.

Supported synchronization methods, sources, and destinations

The following table describes the sources and destinations that are supported by real-time synchronization nodes.

> Note Real-time synchronization nodes do not support a synchronization view.

Synchronization method	Source	Destination	References for configuring data sources	References for configuring synchronization nodes
Synchronize data from a single table in a source	• MySQL Binlog • DataHub • LogHub (SLS) • Kafka • PolarDB • SQL Server	• MaxCompute • Hologres • AnalyticDB MySQL • Elasticsearch • DataHub • Kafka	• Configure a data source (PolarDB) • Configure data sources for data synchronization from MySQL	Configure and manage a real-time data sync node
Synchronize data from all tables in a source	• PolarDB MySQL Note Only PolarDB for MySQL is supported. • Oracle • MySQL	MaxCompute	• Configure a data source (PolarDB) • Configure data sources for data synchronization from Oracle • Configure data sources for data synchronization from MySQL	Configure and manage a real-time data sync node
	• PolarDB MySQL Note Only PolarDB for MySQL is supported. • Oracle • MySQL • SQL Server	Hologres	• Configure a data source (PolarDB) • Configure data sources for data synchronization from Oracle • Configure data sources for data synchronization from MySQL	Configure and manage a real-time sync node

Synchronization method	Source	Destination	References for configuring data sources	References for configuring synchronization nodes
	- PolarDB MySQL Note Only PolarDB for MySQL is supported. - OceanBase - MySQL - Oracle	DataHub	- Configure data sources for data synchronization from PolarDB - Configure a data source (ApsaraDB for OceanBase) - Configure data sources for data synchronization from MySQL - Configure data sources for data synchronization from Oracle	Configure and manage a real-time data synchronization node
	MySQL	Kafka	Configure data sources for data synchronization from MySQL	Configure and manage a real-time data synchronization node

Resource usage and pricing

Before you use a data synchronization node to synchronize data, you must purchase an exclusive resource group for data integration and add the resource group to DataWorks for subsequent use.

The following table describes the performance metrics of exclusive resource groups for Data Integration.

Specifications	Maximum number of parallel threads for a batch synchronization node	Maximum number of parallel real-time synchronization nodes for a single table in a source	Maximum number of parallel real-time synchronization nodes for multiple tables in a source	Maximum number of parallel real-time synchronization nodes for table shards
4c8g	8	3	3	Not supported
8c16g	16	6	6	1
12c24g	24	9	9	1
16c32g	32	12	12	2
24c48g	48	18	18	3

For information about the pricing of exclusive resource groups for Data Integration in different regions, see Pricing. The actual prices on the buy page prevail.

You can estimate the required resources and purchase an exclusive resource group for Data Integration based on the amount of data that you want to synchronize. For more information about exclusive resource groups for Data Integration, see Exclusive resources for Data Integration.

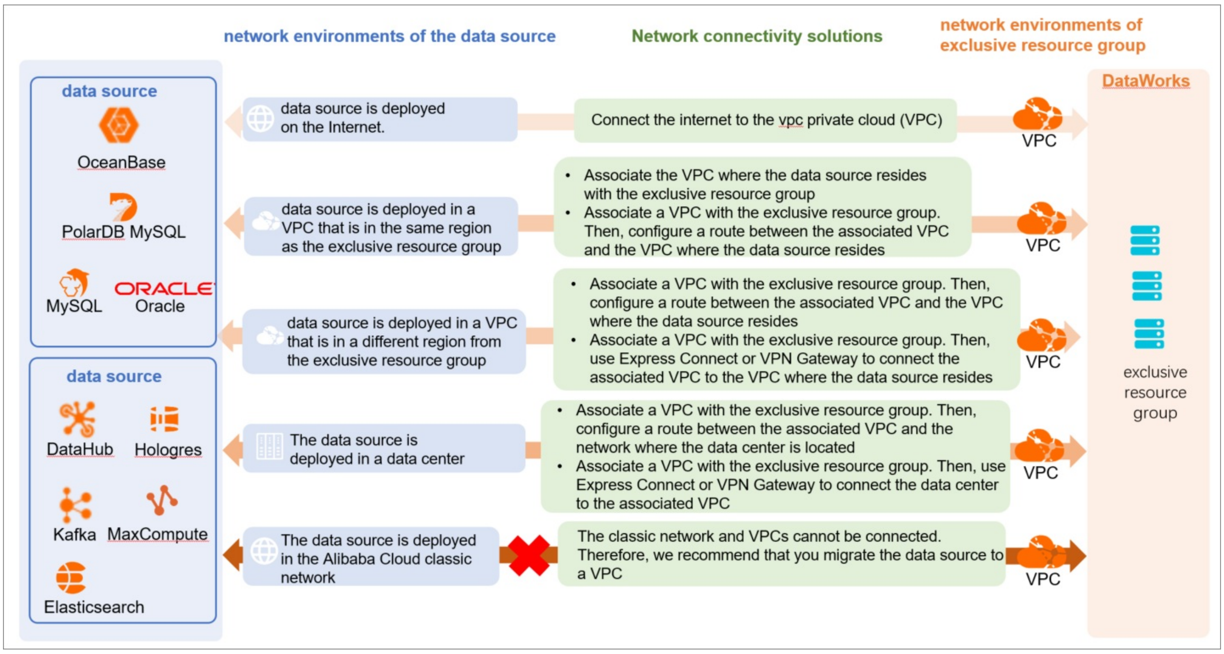
Best practice: We recommend that you use different resource groups for batch synchronization nodes and real-time synchronization nodes. This ensures the isolation of resources used by the two types of nodes, which prevents issues such as resource preemption and runtime exception. Otherwise, CPU resources, memory resources, and networks used by the two types of nodes may affect each other. In this case, batch synchronization nodes may slow down or real-time synchronization nodes may be delayed. Even worse, out of memory (OOM) errors may occur due to the lack of resources.

Network connectivity solutions

For more information about network connectivity solutions, see Overview of network connectivity solutions. This section describes the solutions that can be used to connect a data source to an exclusive resource group.

An exclusive resource group for Data Integration is essentially a group of ECS instances. After you purchase such an exclusive resource group, it is isolated from other services. You must associate the resource group with a virtual private cloud (VPC) to ensure network connectivity between the resource group and data sources during subsequent data synchronization.

The network connectivity solution varies based on the network environments of a source and a destination.



data source is deployed on the Internet.

Connect the internet to the vpc private cloud (VPC)

data source is deployed in a VPC that is in the same region as the exclusive resource group

- Associate the VPC where the data source resides with the exclusive resource group
- Associate a VPC with the exclusive resource group. Then, configure a route between the associated VPC and the VPC where the data source resides

data source is deployed in a VPC that is in a different region from the exclusive resource group

- Associate a VPC with the exclusive resource group. Then, configure a route between the associated VPC and the VPC where the data source resides
- Associate a VPC with the exclusive resource group. Then, use Express Connect or VPN Gateway to connect the associated VPC to the VPC where the data source resides

The data source is deployed in a data center

- Associate a VPC with the exclusive resource group. Then, configure a route between the associated VPC and the network where the data center is located
- Associate a VPC with the exclusive resource group. Then, use Express Connect or VPN Gateway to connect the data center to the associated VPC

The data source is deployed in the Alibaba Cloud classic network

The classic network and VPCs cannot be connected. Therefore, we recommend that you migrate the data source to a VPC

- The data source is deployed on the Internet.

  Connect the data source to the virtual private cloud (VPC) that is associated with the exclusive resource group.
- The data source is deployed in a VPC that is in the same region as the exclusive resource group.
  - Same zone: Associate the exclusive resource group with the VPC in which the data source resides.
  - Different zones: Associate the exclusive resource group with a VPC. Then, configure a route between the associated VPC and the VPC in which the data source resides.
- The data source is deployed in a VPC that is in a different region from the region in which the exclusive resource group resides.
  - Associate the exclusive resource group with a VPC. Then, configure a route between the associated VPC and the VPC in which the data source resides.
  - Associate the exclusive resource group with a VPC. Then, use Express Connect or VPN Gateway to connect the associated VPC to the VPC in which the data source resides.
- The data source is deployed in a data center.
  - Associate the exclusive resource group with a VPC. Then, configure a route between the associated VPC and the network to which the data center is connected.
  - Associate the exclusive resource group with a VPC. Then, use Express Connect or VPN Gateway to connect the network to which the data center is connected to the associated VPC.
- The data source is deployed on the Alibaba Cloud classic network.

  The classic network and VPCs cannot be connected. Therefore, we recommend that you migrate the data source to a VPC.

Procedure

To use a synchronization solution of DataWorks, perform the following steps:

1. Plan and configure resources.

   Estimate the required resources and purchase an exclusive resource group for Data Integration based on the amount of data that you want to synchronize and the network environment. Configure the resources to ensure network connectivity.
2. Configure data sources.

After you establish network connections for data sources between which you want to synchronize data, configure the data sources to ensure accessibility. For example, make sure that the IP addresses of the exclusive resource groups are added to the IP address whitelists of the data sources. Otherwise, the synchronization fails.

3. Add data sources.

   Add the data sources to DataWorks as the source and destination. This way, you can associate the data sources when you create a synchronization solution.

4. Create and configure a synchronization solution.

   Create a synchronization solution and set the parameters based on the synchronization scenario.

> ⓘ **Note** If you set the Table creation method parameter to Create Table when you configure a destination table for the synchronization node, you can click the table name to view and modify the table creation statements. Check whether the table creation statements meet your requirements.

For more information about the synchronization between sources and destinations, see the following topics:

- Synchronize data in a single table
- Synchronize all data from a source to MaxCompute
- Synchronize all data from a source to Hologres
- Synchronize all data from a source to AnalyticDB for MySQL
- Synchronize all data from a source to DataHub

3.2. Plug-ins for data sources that support real-time synchronization

You can use the reader, writer, and conversion plug-ins for various data sources to synchronize data in real time. This topic describes the plug-ins for data sources that support real-time synchronization.

Plug-in type	Plug-in name	References
Reader	MySQL Binlog Reader	Configure MySQL Binlog Reader
	DataHub Reader	Configure DataHub Reader
	LogHub (SLS) Reader	Configure LogHub (SLS) Reader
	Kafka Reader	Configure Kafka Reader
	PolarDB Reader	Configure PolarDB reader
	PostgreSQL Reader	For more information about how to configure a PostgreSQL data source for batch and real-time synchronization, see Configure a source PostgreSQL data source.
Writer	MaxCompute Writer	Configure MaxCompute Writer
	Hologres Writer	Configure Hologres Writer
	AnalyticDB for MySQL Writer	Configure AnalyticDB for MySQL Writer
	DataHub Writer	Datahub writer
	Kafka Writer	Configure Kafka Writer
	Elasticsearch Writer	Configure Elasticsearch Writer
Conversion	Data Filtering	Configure Data Filtering
	String Replace	String replacement

conversion

Plug-in type	Plug-in name	References
	Data Masking	Configure data de-identification

Note

- You cannot run a real-time synchronization node on the node configuration tab. Instead, you must run a real-time synchronization node in the production environment after you save and commit the node.
- Real-time synchronization nodes can be run only on an exclusive resource group for Data Integration. For more information, see DataWorks exclusive resources.

Limits

If you want to synchronize data in real time from a relational database, such as a MySQL, a PolarDB for MySQL, an Oracle, or a PolarDB-X database, note that the following operations are not supported:

- Online DDL operations are not supported. This feature allows you to perform data definition language (DDL) operations on tables, such as creating an index, without blocking data manipulation language (DML) operations and SELECT queries that run concurrently with the DDL operations. In this case, errors and data quality issues occur when data is written to the destination in real-time synchronization.
- If a column is inserted to or deleted from the source between the start time of the real-time synchronization node and the current time, errors and data quality issues occur when data is written to the destination.

3.3. Fields used for real-time synchronization

This topic describes the fields that Data Integration uses to synchronize data in real time.

The following table describes the format of a data record synchronized by Data Integration from a MySQL or Oracle database in real time.

_sequence_id_	_operation_type_	_execute_time_	_before_image_	_after_image_	Field 1	Field 2	Field 3
The position of the data record.	The type of the operation. Valid values: I, D, and U.	The timestamp when the data record was generated.	Indicates whether the data record stores the original data. Valid values: Y and N.	Indicates whether the data record stores the updated data. Valid values: Y and N.	Field 1 in the source database.	Field 2 in the source database.	Field 3 in the source database.

When Data Integration synchronizes data from databases such as MySQL, Oracle, LogHub and PolarDB to DataHub or Kafka in real time, Data Integration adds five fields to data records in the destination data store. These fields are used for operations such as metadata management, sorting, and de-duplication. The following table describes the fields that Data Integration adds to the destination data store.

Field	Type	Description
_sequence_id_	STRING	The position of the synchronized data record in the binary log file. It consists of the name of the binary log file and the offset of the data record.
_operation_type_	STRING	The type of the operation. Valid values: • I: INSERT. • D: DELETE. • U: UPDATE.
_execute_time_	LONG	The timestamp in the binlog file, indicating when the data record was generated.

Field	Type	Description
_before_image_	STRING	Indicates whether the data record stores the original data. Valid values: Y and N.
_after_image_	STRING	Indicates whether the data record stores the updated data. Valid values: Y and N.

In incremental data records generated for the INSERT, UPDATE, and DELETE operations, the _before_image_ and _after_image_ fields are set:

- INSERT: An incremental data record is generated after an INSERT operation. This data record stores the new data. For this data record, the value of _before_image_ is N, whereas the value of _after_image_ is Y.
- UPDATE: Two incremental data records are generated for an UPDATE operation. One stores the original data, and the other stores the updated data. The two data records have the same values for _sequence_id_, _operation_type_, and _execute_time_.

  For the data record that stores the original data, the value of _before_image_ is Y, whereas the value of _after_image_ is N. For the data record that stores the updated data, the value of _before_image_ is N, whereas the value of _after_image_ is Y.
- DELETE: An incremental data record is generated after the DELETE operation. This data record stores the original data. For this data record, the value of _before_image_ is Y, whereas the value of _after_image_ is N.

3.4. Synchronize data in a single table

3.4.1. Plan and configure resources

When you use DataWorks to synchronize data, you can use only exclusive resource groups for data integration to run real-time data synchronization nodes. This topic describes the resources and configurations required to run real-time data synchronization nodes.

Context

- Resource planning and preparation

  Before you use a data synchronization node to synchronize data, you must purchase an exclusive resource group for data integration and add the resource group to DataWorks for subsequent use.

  For more information about exclusive resource groups for Data Integration, see Exclusive resources for Data Integration.
- Network connections

  An exclusive resource group for Data Integration is essentially a group of Elastic Compute Service (ECS) instances. After you purchase and create such an exclusive resource group, it is isolated from other services. You must associate the resource group with a virtual private cloud (VPC) to ensure network connectivity between the resource group and data sources during subsequent data synchronization.

What's next

After you plan and configure resources, you can configure data sources. You must configure network connectivity for the data sources and permissions to access the data sources. If the data sources that you want to use do not need to be configured, you can proceed to the next step to add the data sources. After the data sources are added, you can create a real-time data synchronization node.

- For more information about how to configure a data source, see Configure a data source (PolarDB) and Configure data sources for data synchronization from MySQL.
- For more information about how to add a data source, see Add data sources.

3.4.2. Configure a data source (PolarDB)

Before you synchronize data in a single PolarDB table in real time, you must perform operations in this topic to configure the network environment, IP address whitelist, and permissions for the data source.

Prerequisites

Before you configure a data source, make sure that the following operations are performed:

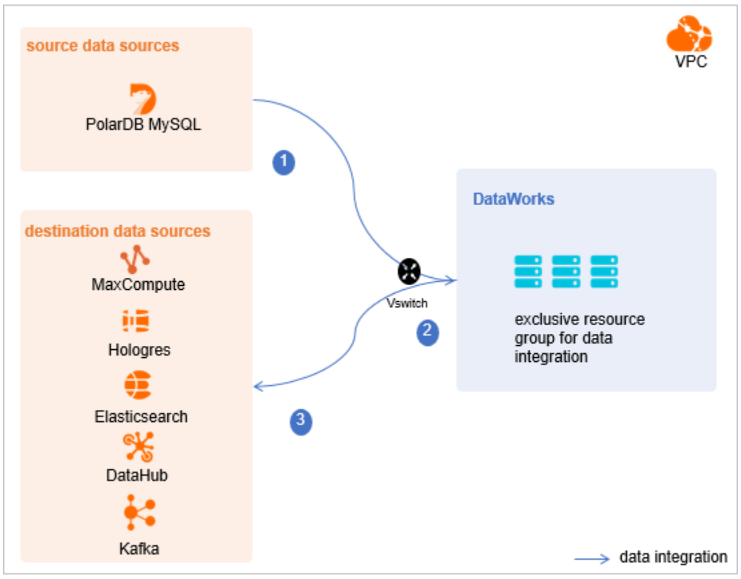
- Prepare data sources: A PolarDB for MySQL cluster and a destination are prepared. The destination can be MaxCompute, Hologres, Elasticsearch, DataHub, or Kafka. In this topic, a PolarDB for MySQL cluster is used as the source.
- Plan and prepare resources: An exclusive resource group for data integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
    - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
    - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data between data sources, make sure that the data sources and the exclusive resource group for data integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

  If the data sources and the exclusive resource group for data integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for data integration to the whitelists of the data sources. This ensures that the exclusive resource group for data integration can be used to access the data sources.

- Create an account and authorize the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source is a PolarDB for MySQL cluster, you must enable the binary logging feature for the cluster. Alibaba Cloud PolarDB for MySQL is fully compatible with MySQL and uses high-level physical logs to replace binary logs. To facilitate the integration between PolarDB and the MySQL ecosystem, you can enable the binary logging feature for PolarDB clusters.

Limits

- Only PolarDB for MySQL clusters can be used as sources in data synchronization solutions. Other types of PolarDB data sources are not supported. In this topic, PolarDB indicates PolarDB for MySQL data sources.
- Only data stored on the primary node of a PolarDB for MySQL cluster can be synchronized.

Procedure

1. Configure a whitelist for the PolarDB for MySQL cluster.

   To add the CIDR block of the VPC where the exclusive resource group for Data Integration resides to a whitelist of the PolarDB for MySQL cluster, perform the following steps:

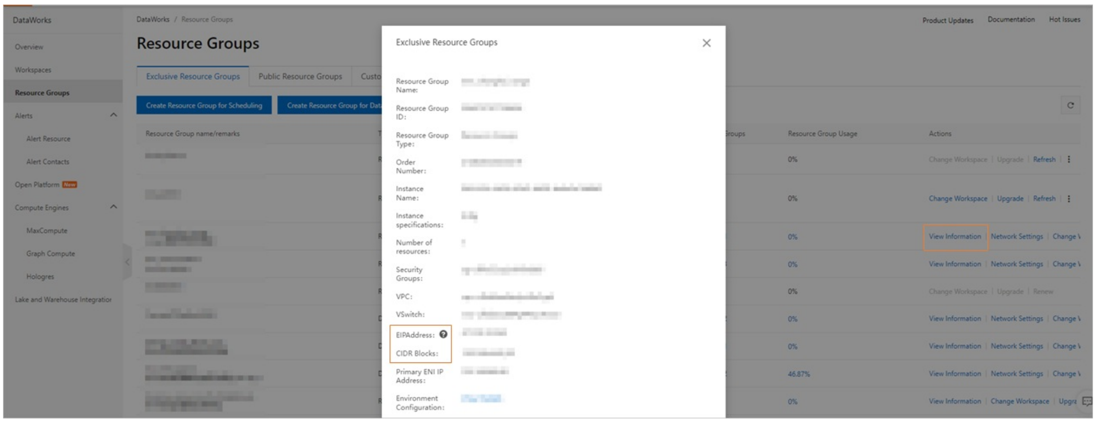
   i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

      a. Log on to the DataWorks console.

      b. In the left-side navigation pane, click **Resource Groups**.

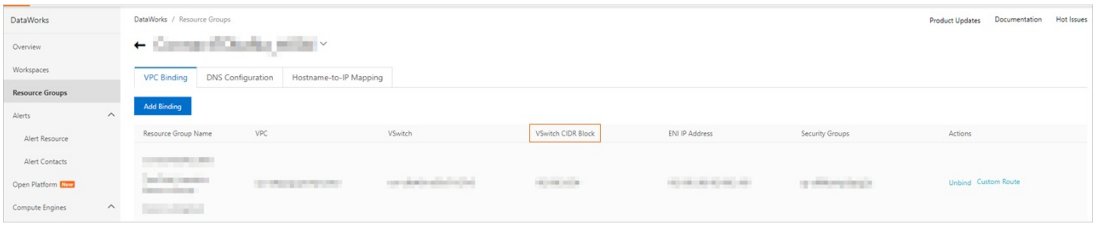
      c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

      d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.

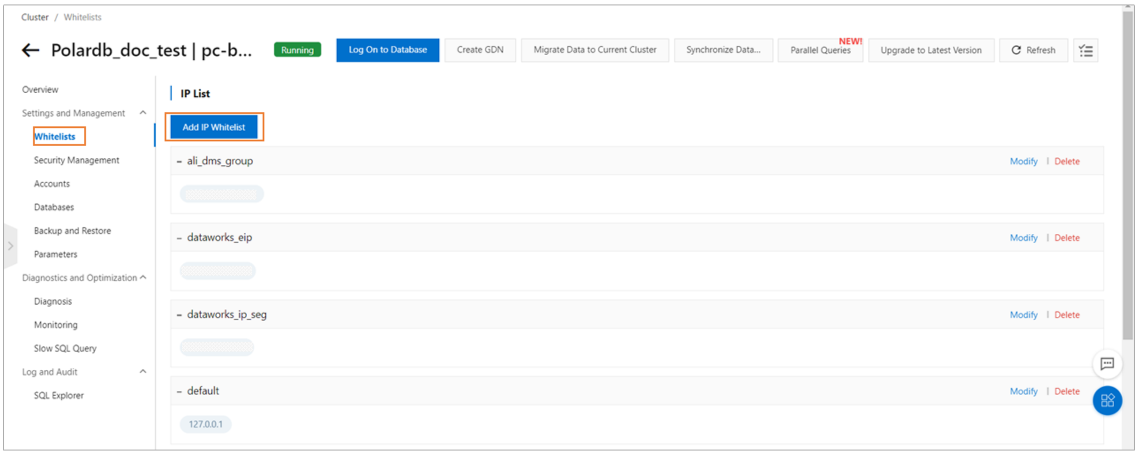


      e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

      f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



   ii. Add the EIP and CIDR blocks recorded in the preceding steps to the whitelist of the PolarDB for MySQL cluster.



   For more information, see Configure an IP whitelist.

2. Create an account and grant the required permissions to the account.

   You must create an account to log on to the database of the PolarDB for MySQL cluster. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

i. Create an account.

For more information, see Create a database account.

ii. Grant the required permissions to the account.

You can run the following command to grant the required permissions to the account, or you can directly assign the `SUPER` role to the account.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Account for data synchronization';
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%';
```

3. Enable the binary logging feature for the PolarDB for MySQL cluster.

For more information, see Enable binary logging.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for data integration are connected, and you can use the authorized account to access the data sources. You can add both the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution. For more information about how to add a data source, see Add data sources.

3.4.3. Configure data sources for data synchronization from MySQL

Before you synchronize data in a single table from MySQL in real time, you can refer to the operations described in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions. You must configure a source MySQL data source and a destination data source.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

- Prepare data sources: A MySQL database and a destination data source are prepared. The destination can be MaxCompute, Hologres, Elasticsearch, DataHub, or Kafka.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.
- Execute the following statement to check the version of your MySQL database. You can use a `MySQL 5.X` or `MySQL 8.X` database.

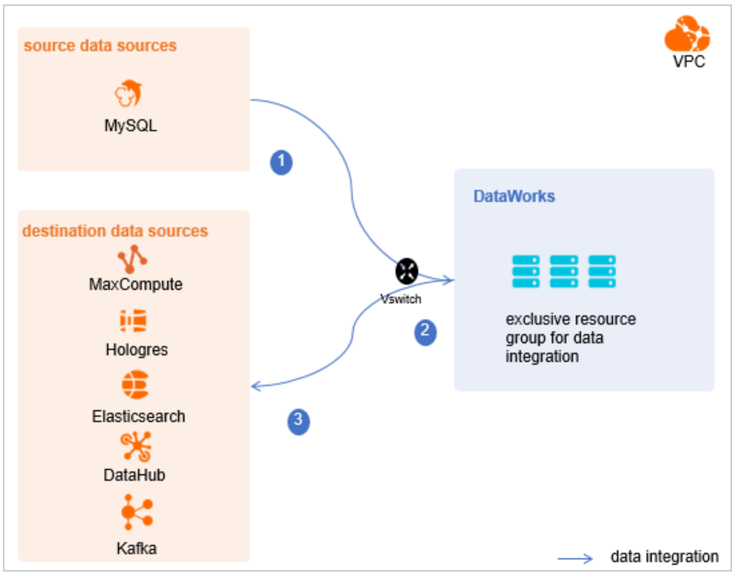
```
select version();
```

> **Note** Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.X` or `V8.X`. PolarDB for MySQL is not supported. If the MySQL version of your ApsaraDB RDS for MySQL database is not `V5.X` or `V8.X`, use an ApsaraDB RDS for MySQL database whose MySQL version is `V5.X` or `V8.X`. Otherwise, the data synchronization node fails to run.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source data source is a MySQL database, you must enable the binary logging feature. Binary logs record changes to all table schemas and modifications on table data. You can execute statements such as CREATE and ALTER to perform operations on table schemas. You can execute statements such as INSERT, UPDATE, and DELETE to perform operations on table data. You can use binary logs to view the change history of the database, back up incremental data and restore data in the database, and replicate data from the primary database to secondary databases.

  Formats of binary logs:

  - Statement: SQL statement-based replication. Binary logs in this format record the SQL statements that are executed to modify data entries.
  - Row: row-based replication. Binary logs in this format record only modification details about data entries in rows.
  - Mixed: replication in mixed mode. This mode combines the statement and row formats. In most cases, binary logs in the statement format are used to record the SQL statements that are executed to modify data entries, such as functions. If data replication from the primary database to secondary databases cannot be implemented by using binary logs in this format, switch to the row format. MySQL determines which format to use based on each SQL statement that is executed.

Limits

Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.X` or `V8.X`. PolarDB for MySQL is not supported.

Procedure

1. Configure a whitelist for the MySQL database.

   Add the CIDR block of the VPC in which the exclusive resource group resides to the whitelist of the MySQL database.

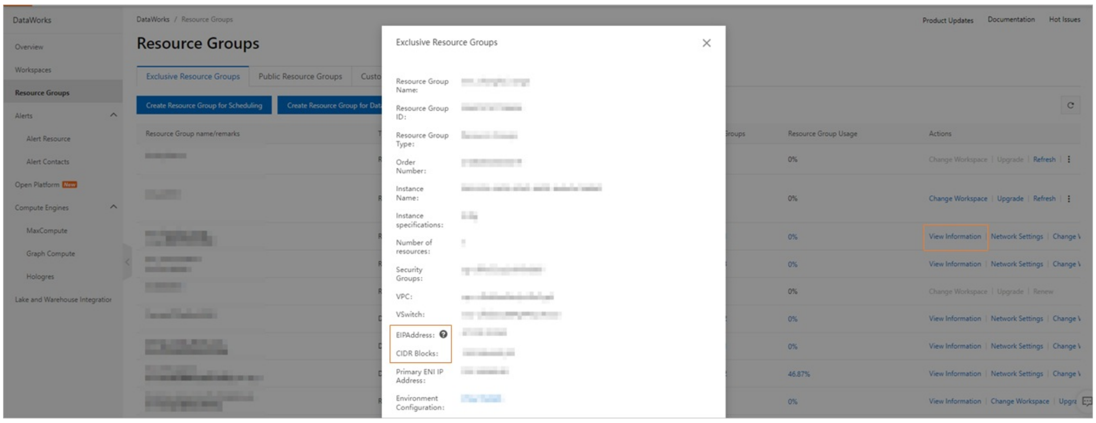
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

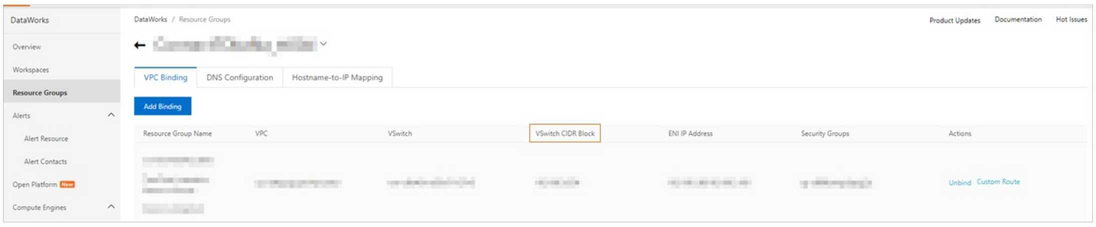
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR block of the exclusive resource group to the whitelist of the MySQL database.

2. Create an account and grant permissions to the account.

You must create an account to log on to the MySQL database. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

i. Create an account.

For more information, see Create an account to access a MySQL database.

ii. Grant permissions to the account.

You can run the following command to grant permissions to the account. Alternatively, you can grant the `SUPER` permission to the account. Replace `Account for data synchronization` with the created account when you execute the specific statement.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Password'; // Create an account that is used to synchronize data and set a password so that you can use the account and password to access the database from a host. % indicates a host.
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%'; // Grant the SELECT, REPLICATION SLAVE, and REPLICATION CLIENT permissions to the account.
```

`*.*` indicates that the synchronization account is granted the preceding permissions on all tables in all databases. You can also grant the preceding permissions on specified tables in the specified database to the synchronization account. For example, to grant the account the preceding permissions on the *user* table in the *test* database, execute the following statement: `GRANT SELECT, REPLICATION CLIENT ON test.user TO 'Account for data synchronization'@'%';` .

> **Note** The `REPLICATION SLAVE` permission is a global permission. You cannot grant this permission on specified tables in the specified database to the synchronization account.

3. Enable the binary logging feature for the MySQL database.

   Perform the following steps to check whether the binary logging feature is enabled and to query the format of binary logs:

   - Execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_bin";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled.

   - If you use a secondary database to synchronize data, execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_slave_updates";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled for the secondary database.

   If the returned result is different from the preceding result, follow the instructions described in the *MySQL documentation* to enable the binary logging feature.

   Execute the following statement to view the format of binary logs:

   ```
   show variables like "binlog_format";
   ```

   Returned result:

   - *ROW*: The format of binary logs is row.
   - *STATEMENT*: The format of binary logs is statement.
   - *MIXED*: The format of binary logs is mixed.

What's next

After the data sources are configured, the source data source, destination data source, and exclusive resource group for Data Integration are connected, and you can use the authorized account to access the data sources. You can add the source data source and destination data source to DataWorks, and associate them with a data sync solution when you create the solution. For more information about how to add a data source, see Add data sources.

3.4.4. Add data sources

Before you configure a node to synchronize data between data sources, you must add the data sources to DataWorks. This way, you can configure the data sources as the source and destination of the node in subsequent operations.

Prerequisites

Before you add a data source, make sure that the following operations are performed:

- Prepare data sources: A source and a destination are prepared.
- Create and authorize an account: An account that is used to access the data sources is created.

Precautions

DataWorks provides workspaces in basic mode and standard mode. A workspace in basic mode does not isolate the development environment from the production environment. A workspace in standard mode isolates the development environment from the production environment.

If you use a workspace in standard mode, you must separately add data sources to the development environment and production environment.

Add a data source (source)

To add a source, you must configure information such as the network connection type and the access account and password as planned. The following topics describe the data sources that can be used as the sources for real-time single-table synchronization and the configurations required for these data sources:

- Add a MySQL data source
- Add a DataHub data source
- Add a LogHub (SLS) data source
- Add a PolarDB data source
- Add an SQL Server data source

Add a data source (destination)

The following topics describe the data sources that can be used as the destinations for real-time single-table synchronization and the configurations required for these data sources:

- Add a MaxCompute data source
- Add a Hologres data source
- Add an Elasticsearch data source
- Add a DataHub data source
- Add an Elasticsearch data source

What's next

After you add data sources, you can create and run a data synchronization node to synchronize data from the source to the destination. For more information, see Configure and manage a real-time data sync node.

3.4.5. Configure and manage a real-time data sync node

After you configure network environments and resources and add data sources to DataWorks, you can create and run a real-time data sync node to synchronize data between the data sources. This topic describes how to create a real-time data sync node and view the status of the node.

Prerequisites

Before you create a real-time data sync node, make sure that the following operations are performed:

- Plan and configure resources
- Configure a data source (PolarDB)
- Configure data sources for data synchronization from MySQL
- Add data sources

Limits

- DataWorks allows you to use only exclusive resource groups for Data Integration to run real-time data sync nodes.
- Real-time data sync nodes support the following data sources and data conversion methods:
  - Sources: MySQL Binlog, DataHub, LogHub, Kafka, PolarDB, and SQL Server
  - Destinations: MaxCompute, Hologres, Elasticsearch, DataHub, and Kafka
  - Data conversion methods: data filtering, string replacement, and data de-identification

- The following rules must be observed when you configure a real-time data sync node:
    - You can synchronize data from one or more source tables to a single destination table. If you want to synchronize data to multiple destination tables, you must create a real-time data sync node for each destination table.
    - You can synchronize data from multiple tables to a single destination table only when the source is MySQL Binlog or SQL Server. The types and schemas of the source tables must be the same. For example, all the source tables are MySQL Binlog tables.

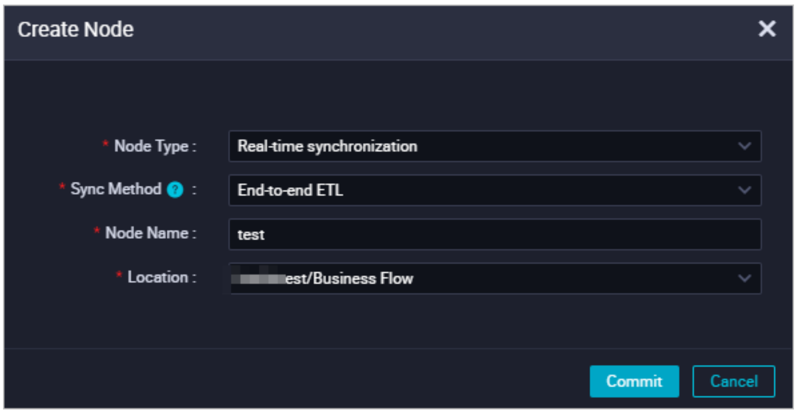
Create a real-time data sync node

1. Log on to the DataWorks console.
2. In the left-side navigation pane, click **Workspaces**.
3. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
4.
5. Create a real-time data sync node.

    i. On the DataStudio page, move the pointer over the +Create icon and choose **Data Integration > Real-time synchronization**.

    Alternatively, find the workflow in which you want to create a real-time data sync node and right-click the **Data Integration**. From the shortcut menu, choose **Create > Real-time synchronization**.

    ii. In the **Create Node** dialog box, set the parameters.



Create Node

* Node Type : Real-time synchronization

* Sync Method : End-to-end ETL

* Node Name : test

* Location : [redacted]est/Business Flow

Parameter	Description
Node Type	The type of the node. Default value: **Real-time synchronization**.
Sync Method	Set this parameter to **End-to-end ETL**. This method is used to synchronize data from one or more source tables to a destination table in real time. You can convert data types during synchronization. Note ▪ You can synchronize data from one or more source tables to a single destination table. If you want to synchronize data to multiple destination tables, you must create a real-time data sync node for each destination table. ▪ You can synchronize data from multiple tables to a single destination table only when the source is MySQL Binlog or SQL Server. The types and schemas of the source tables must be the same. For example, all the source tables are MySQL Binlog tables.
Node Name	The name of the node. The name must be 1 to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).
Location	The folder in which the real-time data sync node is stored.

   iii. Click **Commit**. You are navigated to the configuration tab of the real-time data sync node.

6. Select a resource group.

   i. On the right side of the configuration tab, click the **Basic Configuration** tab.

   ii. In the panel that appears, select the resource group that you want to use from the **Resource Group** drop-down list.

   > ⓘ **Note**
   >
   > DataWorks allows you to use only exclusive resource groups for Data Integration to run real-time data sync nodes.
   >
   > If no exclusive resource group for Data Integration exists, click **Create Exclusive Resource Group for Data Integration** to create a resource group. For more information, see Overview.

7. Configure the source data source.

   i. From the left-side **Input** section of the configuration tab of the real-time data sync node, drag the required source component to the configuration canvas on the right.

   ii. Click the component and set the parameters in the panel that appears.

   The following topics describe the data sources that can be used as the sources for real-time single-table synchronization and the configurations required for these data sources:

   - Configure MySQL Reader
   - Datahub reader
   - LogHub (SLS) reader
   - Configure Kafka Reader
   - ApsaraDB for PolarDB reader

8. (Optional)Configure a data conversion method.

   If you want to convert data types during synchronization, you can configure a data conversion method.

   i. From the left-side **Conversion** section of the configuration tab of the real-time data sync node, drag the required data conversion method component to the configuration canvas on the right.

   ii. Click the component and set the parameters in the panel that appears.

   The following topics describe the supported data conversion methods and the configurations required for these methods:

   - Configure Data Filtering
   - String replacement
   - Configure data de-identification

9. Configure the destination data source.

   i. From the left-side **Output** section of the configuration tab of the real-time synchronization node, drag the required destination component to the configuration canvas on the right.

   ii. Click the component and set the parameters in the panel that appears.

   The following topics describe the data sources that can be used as the destinations for real-time single-table synchronization and the configurations required for these data sources:

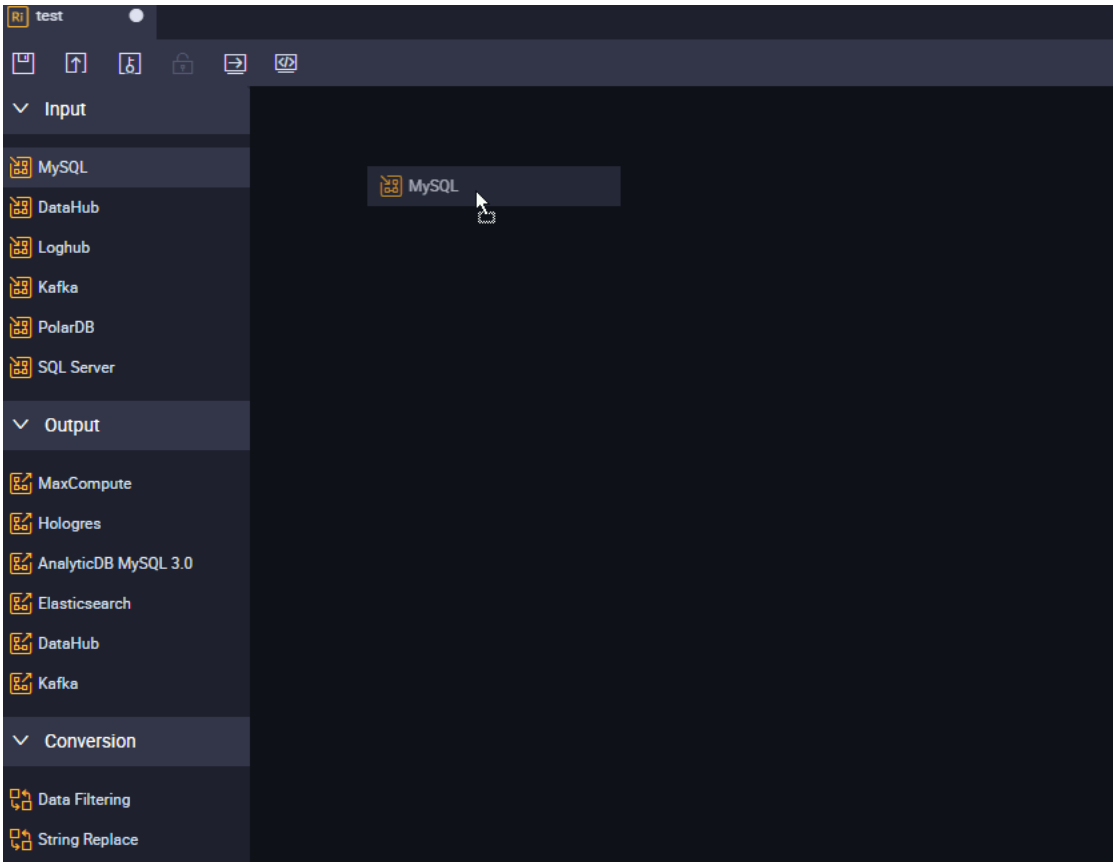
   - Configure MaxCompute Writer
   - Configure Hologres Writer
   - Datahub writer
   - Configure Kafka Writer
   - Configure Elasticsearch Writer

10. Connect the source component to the destination component.

    After the source and destination components are added, you can connect the components by drawing lines. This way, the components can synchronize data based on the connection.

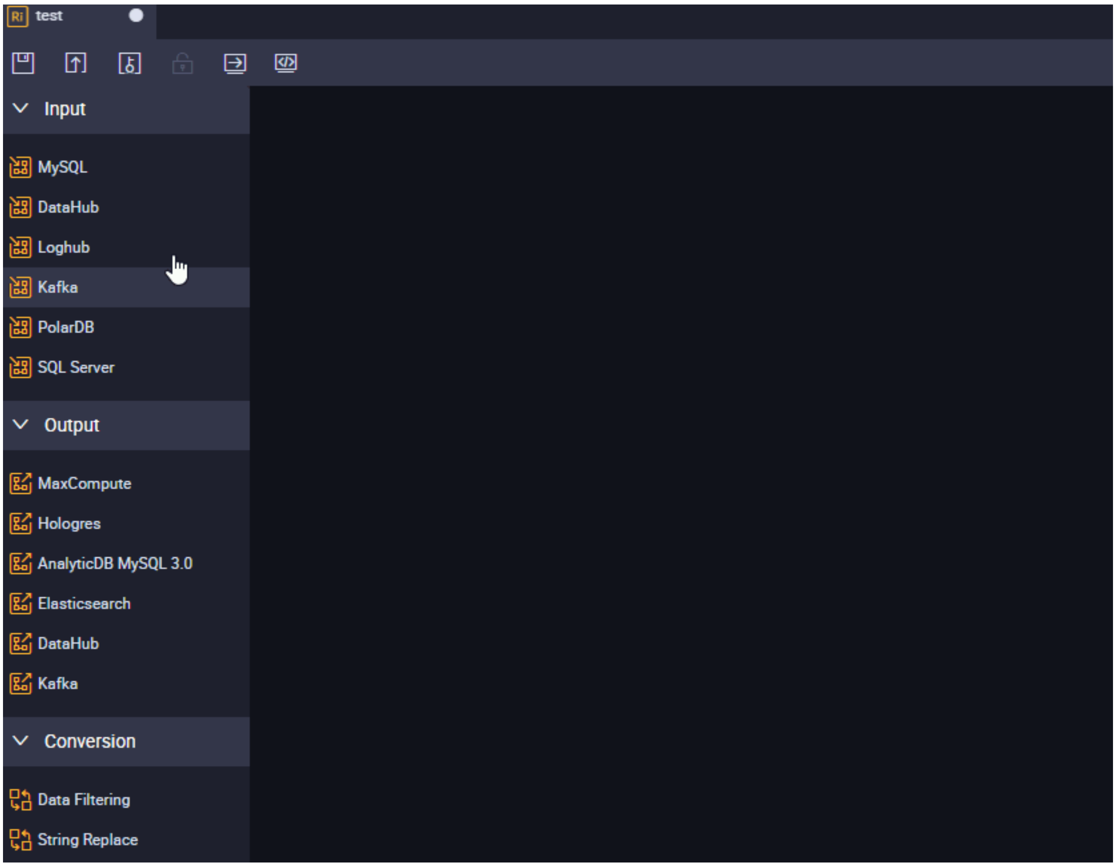
    - Example 1: The following figure shows the process during which data is synchronized from MySQL Binlog to MaxCompute in real time. During the synchronization, data types are not converted.

- Source: MySQL Binlog
- Destination: MaxCompute
- Direction of data synchronization: The source component is connected to the destination component. Data is synchronized from MySQL Binlog to MaxCompute.



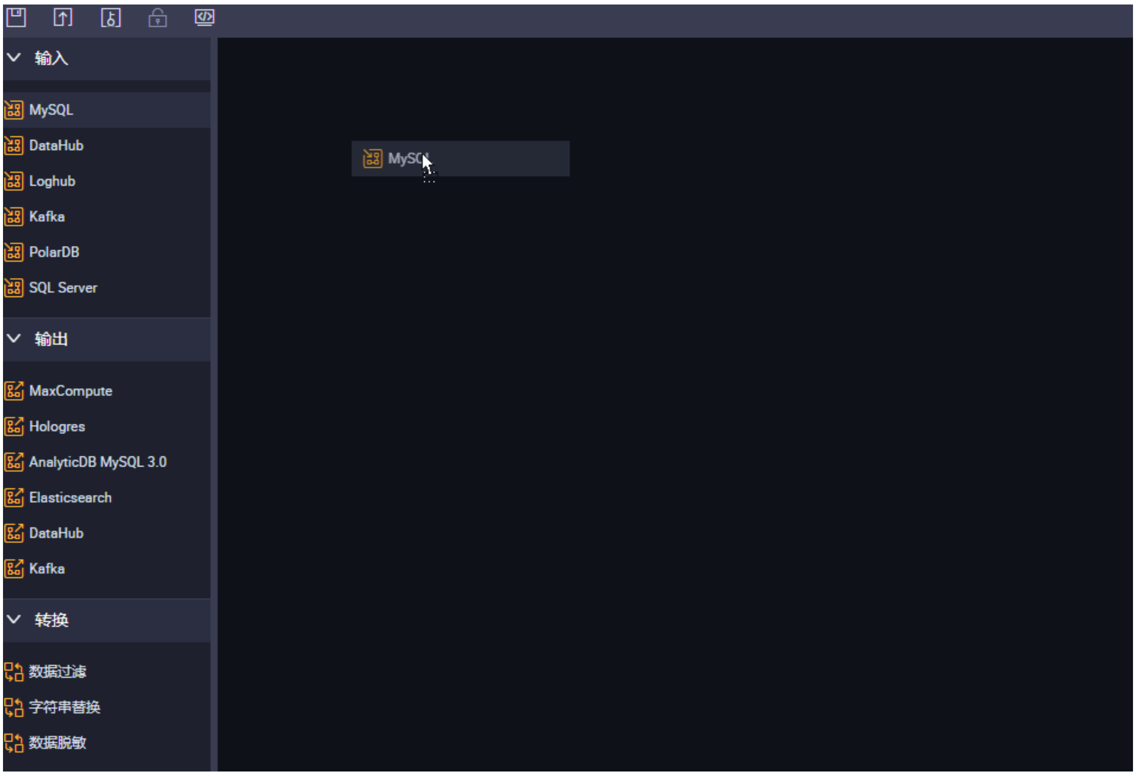
- Example 2: The following figure shows the process during which data is synchronized from MySQL Binlog to MaxCompute in real time. During the synchronization, data types are converted.
  - Source: MySQL Binlog
  - Data conversion method: data filtering. The data in the source is filtered by using this method.
  - Destination: MaxCompute

- Direction of data synchronization: The source component is connected to the data filtering component, and the data filtering component is connected to the destination component. Data that is read from MySQL Binlog is filtered. Then, the filtered data is synchronized to MaxCompute.



- Example 3: The following figure shows the process during which data is synchronized from MySQL to MaxCompute in real time. During the synchronization, data is de-identified.
  - Source: MySQL
  - Data conversion method: data de-identification. The data in the source is de-identified by using this method.
  - Destination: MaxCompute

- Direction of data synchronization: The source component is connected to the data masking component, and the data de-identification component is connected to the destination component. Data that is read from MySQL is de-identified. Then, the de-identified data is synchronized to MaxCompute.



Commit and deploy the real-time data sync node

1.

Run the real-time data synchronization node

1. Go to the Operation Center page.

   After you commit and deploy the real-time data sync node, click **Operation Center** in the upper-right corner of the DataStudio page to manage the node on the **Real Time DI** page.

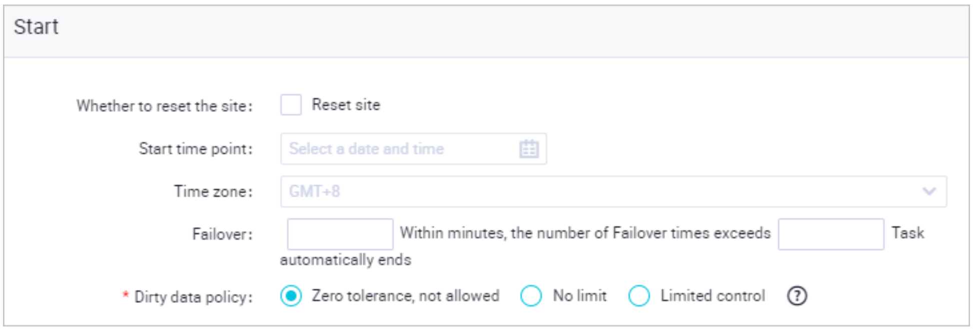
2. View the details of a real-time data sync node.

   On the **Real Time DI** page, find the real-time data sync node that you want to view and click the node name.

3. Run the real-time data synchronization node.

   i. Go to the Real Time DI page of Operation Center, find the real-time data synchronization node that you created and click **Start** in the **Operation** column.

ii. In the **Start** dialog box, configure the parameters.



Start

Whether to reset the site: □ Reset site

Start time point: Select a date and time

Time zone: GMT+8

Failover: [ ] Within minutes, the number of Failover times exceeds [ ] Task automatically ends

* Dirty data policy: ● Zero tolerance, not allowed ○ No limit ○ Limited control ⓘ

Parameter	Description
Whether to Set Start Offset	Specifies whether to set the point in time for the next startup. If you select Reset site, the **Start time point** and **Time zone** parameters are required.
Time	The date and time for starting the real-time data synchronization node.
Time zone	The time zone in which the real-time data synchronization node is run. You can select a time zone from the **Time zone** drop-down list.
Failover	The maximum number of failovers allowed within the specified time range. ⓘ **Note** If you do not configure this parameter, the system automatically stops the node if the number of failovers exceeds 100 within 5 minutes. This prevents excessive resource consumption caused by the frequent starting of the node.
Dirty data policy	▪ **Zero tolerance, not allowed**: The real-time data synchronization node is automatically stopped if dirty data is generated during data synchronization. ▪ **No limit**: The real-time data synchronization node can normally run regardless of whether dirty data is generated during data synchronization. ▪ **Limited control**: The real-time data synchronization node is automatically stopped if the amount of dirty data that is generated during data synchronization exceeds a specified value.

iii. Click **Confirm**.

Manage the real-time data sync node

- Stop a real-time data sync node that is running.

  Find the real-time data sync node that you want to stop and click **Stop** in the Operation column. In the message that appears, click **Stop**.

- Undeploy a real-time data sync node that is not running.

  Find the real-time data sync node that you want to undeploy and click **Undeploy** in the Operation column. In the message that appears, click **Undeploy**.

- View the alert information of a real-time data sync node.

  Find the real-time data sync node that you want to view and click **Alert settings** in the Actions column. In the **Alert settings** dialog box, view the alert events and alert rules.

- Configure alert rules for a real-time data sync node.

  i. Find the real-time data sync node for which you want to configure alert rules and click **Configure Alert Rule** in the lower part of the **Real Time DI** page.

  ii. In the **New rule** dialog box, set the parameters that are described in the following table.

Parameter	Description
Name	The name of the alert rule.
Description	The description of the alert rule.
Indicators	The metric for which an alert is reported. Valid values: ▪ **Status** ▪ **Business delay** ▪ **Failover** ▪ **Dirty Data** ▪ **Not Supported by DDL Statement**
Threshold	The threshold for reporting an alert. Specify the **WARNING In** and **CRITICAL In** parameters. The default values of the parameters are 5 minutes.
Alarm interval	The interval at which an alert is reported. The default value is 5 minutes.
WARNING	The method that is used to send alert notifications. You can specify one or more methods. Valid values: **Mail**, **SMS**, and **DingTalk**. **Note** Only Singapore, Malaysia(Kuala Limpur), and Germany(Frankfurt) support the SMS reminding method. To use the SMS reminding method in other regions, submit a ticket to contact DataWorks technical support.
CRITICAL	
Receiver (Non-DingTalk)	The recipient of alert notifications.

iii. Click **Confirm**.

- Modifies alert rules for real-time data sync nodes at a time.
  i. Select one or more real-time data sync nodes for which you want to modify alert rules and click **Operation alarm** in the lower part of the **Real Time DI** page.
  ii. In the **Operation alarm** dialog box, modify the values of the **Type** and **Indicators** parameters.
  iii. Click **Confirm**.

3.4.6. Reader

3.4.6.1. Configure MySQL Reader

You can use MySQL Reader to read data from tables in your MySQL database in real time by subscribing to binary logs (binlogs). This topic describes how to configure MySQL Reader. This topic also describes the network environment and permissions that you must prepare before you configure MySQL Reader.

Prerequisites

Before you configure MySQL Reader, make sure that the following operations are performed:

- Prepare a data source: A MySQL data source is created.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you add the data source to DataWorks, connect the data source to an exclusive resource group for Data Integration based on your business requirements. After you connect the data source to the exclusive resource group for Data Integration, configure network access settings such as a vSwitch and a whitelist.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

  For more information about how to configure a whitelist, see Configure whitelists for data sources.

- Execute the following statement to check the version of your MySQL database. You can use a `MySQL 5.X` or `MySQL 8.X` database.

  ```
  select version();
  ```

  > ⓘ **Note** Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.X` or `V8.X`. PolarDB for MySQL is not supported. If the MySQL version of your ApsaraDB RDS for MySQL database is not `V5.X` or `V8.X`, use an ApsaraDB RDS for MySQL database whose MySQL version is `V5.X` or `V8.X`. Otherwise, the data synchronization node fails to run.

- Prepare an account and grant permissions to the account.

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

  For more information, see Create and authorize an account.

- Enable the MySQL binary logging feature. This feature is required only for real-time data synchronization. For more information about real-time data synchronization, see Overview.

  If the source data source is a MySQL database, you must enable the binary logging feature. Binary logs record changes to all table schemas and modifications on table data. You can execute statements such as CREATE and ALTER to perform operations on table schemas. You can execute statements such as INSERT, UPDATE, and DELETE to perform operations on table data. You can use binary logs to view the change history of the database, back up incremental data and restore data in the database, and replicate data from the primary database to secondary databases.

  Formats of binary logs:

  - Statement: SQL statement-based replication. Binary logs in this format record the SQL statements that are executed to modify data entries.
  - Row: row-based replication. Binary logs in this format record only modification details about data entries in rows.
  - Mixed: replication in mixed mode. This mode combines the statement and row formats. In most cases, binary logs in the statement format are used to record the SQL statements that are executed to modify data entries, such as functions. If data replication from the primary database to secondary databases cannot be implemented by using binary logs in this format, switch to the row format. MySQL determines which format to use based on each SQL statement that is executed.

  For more information about how to enable the binary logging feature, see Enable the binary logging feature.

Limits

- Data Integration does not support data synchronization from a read-only database by using MySQL Reader.
- MySQL Reader reads data in your MySQL database in real time by subscribing to binlogs. MySQL Reader supports data synchronization only from ApsaraDB RDS for MySQL `5.x` and `8.x`. MySQL Reader does not support data synchronization from Distributed Relational Database Service (DRDS).

Procedure

1. Go to the **DataStudio** page.

   i. Log on to the DataWorks console.

   ii. In the left-side navigation pane, click **Workspaces**.

   iii. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.

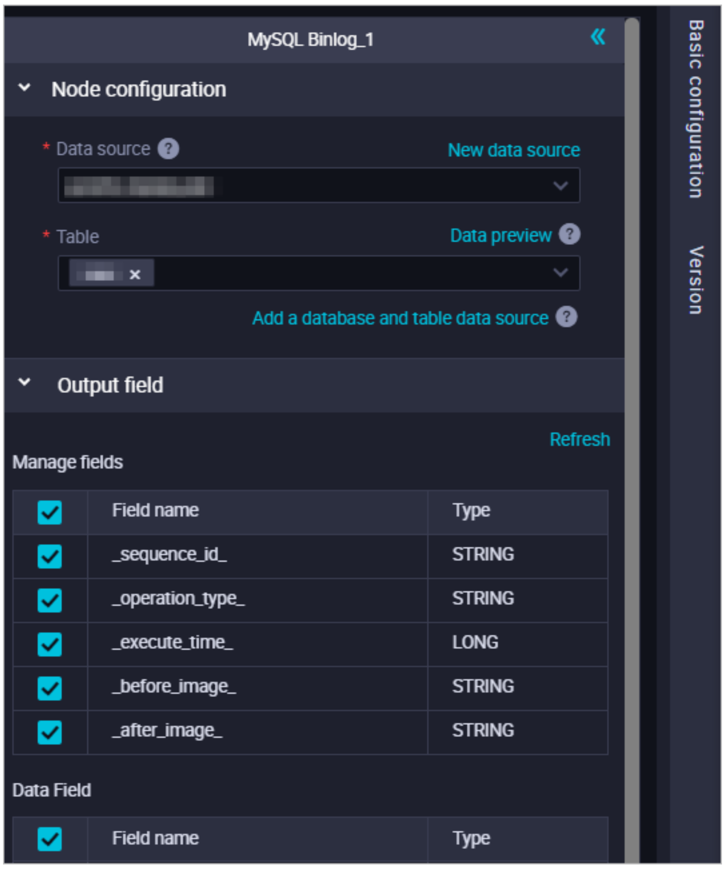
2. In the Scheduled Workflow pane, move the pointer over the +Create icon and choose **Data Integration > Real-time synchronization**.

   Alternatively, click the required workflow, right-click **Data Integration**, and then choose **Create > Real-time synchronization**.

3. In the **Create Node** dialog box, set the Sync Method parameter to **End-to-end ETL** and set the **Node Name** and **Location** parameters.

> Notice The node name can be up to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).

4. Click **Commit**.
5. On the configuration tab of the real-time sync node, choose **Input > MySQL** and drag MySQL to the canvas on the right.
6. Click the **MySQL** node. In the **Node Configuration** panel, set the parameters.



MySQL Binlog_1

Node configuration

* Data source ? New data source

[redacted]

* Table Data preview ?

[redacted] ×

Add a database and table data source ?

Output field

Refresh

Manage fields

☑	Field name	Type
☑	_sequence_id_	STRING
☑	_operation_type_	STRING
☑	_execute_time_	LONG
☑	_before_image_	STRING
☑	_after_image_	STRING

Data Field

☑	Field name	Type

Parameter	Description
Data source	The data source. You can only select a MySQL data source. If no data source is available, click **New data source** to add one on the **Data Source** page.
Table	The name of the table from which you want to read data. You can click **Preview Data** on the right to preview the selected table. In the database sharding scenario, MySQL Reader can read data from multiple tables and databases in real time. Notice To prevent errors, make sure that the tables use the same schema.

Parameter	Description
Output Fields	The fields that you want to synchronize. You can select management fields and data fields. ○ **Manage fields**: the additional fields that are automatically added when the fields in the source tables are synchronized to the destination. These fields facilitate data management, sorting, and deduplication. ○ **Data fields**: the fields in the source tables that you want to synchronize. For more information, see Fields used for real-time synchronization.

MySQL Reader supports the sharding feature. To use this feature, click **Add a database and table data source** and select the data sources and tables from the drop-down lists to synchronize data from multiple tables and databases.

> **Notice** To prevent errors, make sure that the selected tables use the same schema.

7. Click the 💾 icon in the top toolbar.

3.4.6.2. Datahub reader

A Datahub reader reads data from Datahub in real time by using the Datahub SDK.

The reader keeps running after it is started and reads data from Datahub when Datahub stores new data. A Datahub reader has the following two features:

- Reads data in real time.
- Reads data concurrently based on the number of shards in Datahub.

Create a Datahub reader

1. Log on to the DataWorks console. In the left-side navigation pane, click Workspaces. On the Workspaces page, find the target workspace and click **Data Analytics** in the Actions column.
2. On the Data Analytics tab, move the pointer over the +Create icon and choose **Data Integration > Real-Time Sync**.

   You can also find the target workflow, right-click **Data Integration**, and choose **Create > Real-Time Sync**.
3. In the **Create Node** dialog box that appears, set **Node Name** and **Location**, and then click **Commit**.
4. On the configuration tab of the real-time sync node, drag **DataHub** under **Reader** to the editing panel.
5. Click the **Datahub reader** node and set parameters in the **Node Settings** section.

Parameter	Description
Connection	The connection to Datahub. In this example, you can only select a Datahub connection. If no connection is available, click **Add Connection** on the right to create one on the **Workspace Manage > Data Source** page.
Topic	The name of the topic from which data is read in Datahub. You can click **Preview** on the right to preview the selected topic.
Start Offset	The start time of the sync node.
Time Zone	The time zone where Datahub resides.
Output Fields	The fields from which data is read.

6. Click 💾 in the toolbar.

3.4.6.3. LogHub (SLS) reader

A LogHub (SLS) reader reads data from LogHub (SLS) topics you specified in real time and supports shard merge and split.

> **Note** After shards are merged or split, duplicate data records may exist but no data will be lost.

Create a LogHub (SLS) reader

1. Log on to the DataWorks console. In the left-side navigation pane, click Workspaces. On the Workspaces page, find the target workspace and click **Data Analytics** in the Actions column.
2. On the Data Analytics tab, move the pointer over the +Create icon and choose **Data Integration > Real-Time Sync**.

   You can also find the target workflow, right-click **Data Integration**, and choose **Create > Real-Time Sync**.
3. In the **Create Node** dialog box that appears, set **Node Name** and **Location**, and then click **Commit**.
4. On the configuration tab of the real-time sync node, drag **LogHub** under **Reader** to the editing panel.
5. Click the **LogHub** node and set parameters in the **Node Settings** section.

Parameter	Description
Connection	The connection to LogHub (SLS). In this example, you can only select a LogHub connection. If no connection is available, click **Add Connection** on the right to create one on the **Workspace Manage > Data Source** page.
Logstore	The name of the Logstore from which data is read in LogHub (SLS). You can click **Preview** on the right to preview the selected Logstore.
Start Offset	The start time of the sync node.
Time Zone	The time zone where LogHub (SLS) resides.
Advanced Settings	Specifies whether to split data in the Logstore.
Output Fields	The fields from which data is read.

6. Click 💾 in the toolbar.

3.4.6.4. Configure Kafka Reader

Kafka Reader uses a Kafka SDK to read data from Kafka in real time.

Context

> **Note**
>
> - Kafka Reader can be used to read data from Message Queue for Apache Kafka data sources and self-managed Kafka data sources. However, the versions of self-managed Kafka data sources must range from 0.10.2 to 2.2.x.
> - Self-managed Kafka data sources whose versions are earlier than 0.10.2 do not support the query of offsets of partition data and do not support timestamps. If you use such a Kafka data source in a synchronization node, the latency data of the synchronization node that is displayed in Operation Center may be incorrect, and the offset from which incremental data starts to be synchronized cannot be reset.

If you want to use the simple authentication and security layer (SASL) authentication mode, contact the technical support and provide the technical support with the SSL root certificate and SASL authentication file. Then, the technical support configures the SSL root certificate and SASL authentication file for the environment in which your synchronization node runs. In addition, you must add the following configurations to the settings of the Kafka data source that is used for your synchronization node:

```
{ "java.security.auth.login.config": "/home/admin/kafka_client_jaas.conf",
 "ssl.truststore.location": "/home/admin/kafka.client.truststore.jks",
 "ssl.truststore.password": "KafkaOnsClient", "security.protocol": "SASL_SSL",
 "sasl.mechanism": "PLAIN", "ssl.endpoint.identification.algorithm": "" }
```

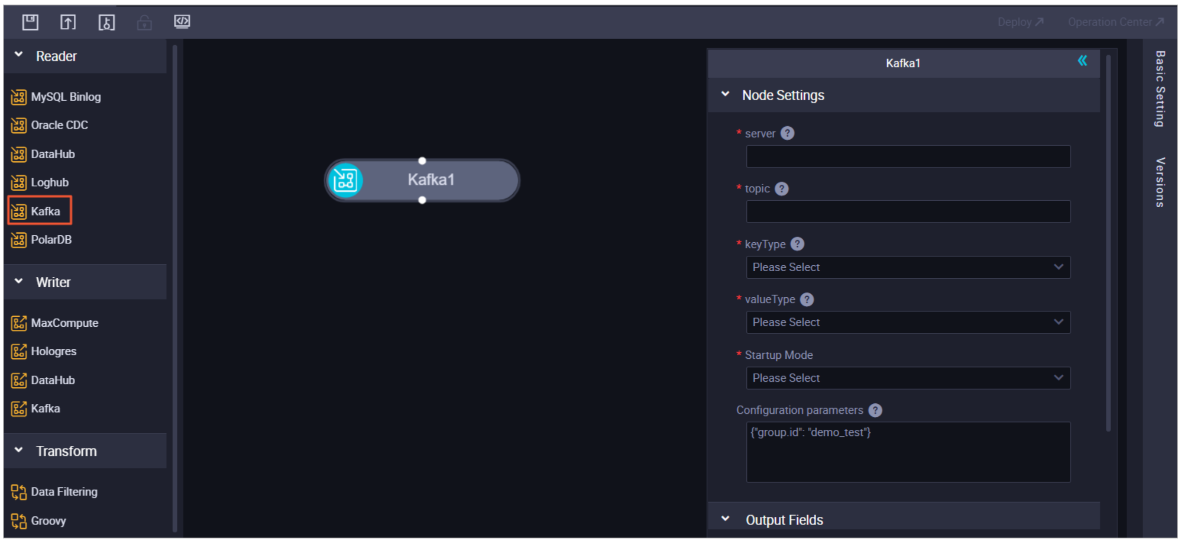
For more information about how to add a Kafka data source, see Add a Kafka data source.

Procedure

1. Go to the **DataStudio** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
2. In the Scheduled Workflow pane, move the pointer over the +Create icon and choose **Data Integration > Real-time synchronization**.

   Alternatively, click the required workflow, right-click **Data Integration**, and then choose **Create > Real-time synchronization**.
3. In the **Create Node** dialog box, set the Sync Method parameter to **End-to-end ETL** and set the **Node Name** and **Location** parameters.

   > **Notice** The node name can be up to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).
4. Click **Commit**.
5. On the configuration tab of the real-time synchronization node, drag **Kafka** in the **Input** section to the canvas on the right.
6. Click the **Kafka** node. In the panel that appears, configure the parameters.



Configuration parameters: {"group.id": "demo_test"}

Parameter	Description
Kafka Cluster Address	The address of the Kafka broker. Specify the address in the `IP address:Port number` format.

Parameter	Description
Topic	The name of the Kafka topic from which you want to read data. Topics are categories in which Kafka maintains the feeds of messages. Each message that is published to a Kafka cluster is assigned to a topic. Each topic contains a group of messages. **Note** Kafka Reader in each synchronization node can read data from only one topic.
key Type	The data type of the keys in the Kafka topic.
Value Type	The data type of the values in the Kafka topic.
Initiation Site	The offset of the first message from which Kafka Reader reads data when the synchronization node runs. ○ EARLIEST: the earliest offset in each partition ○ LATEST: the latest offset in each partition ○ TIMESTAMP: the start offset that is specified in Operation Center ○ GROUP_OFFSETS: the offset of the message that is previously consumed by the consumer group specified by the group.id parameter **Note** If you rerun the node, the Initiation Site parameter does not take effect. You can specify a new start offset in Operation Center to rerun the node. If you do not specify a new start offset in Operation Center, Kafka Reader reads messages from the buffer offset.
Output Mode	The mode in which Kafka Reader parses messages in the Kafka topic. Valid values: ○ Single-row Output: Kafka Reader parses messages as unstructured strings or JSON objects. One message is parsed into one output record. ○ Multi-row Output: Kafka Reader parses messages as JSON arrays. Each array element is parsed into one output record. Therefore, one message may be parsed into multiple output records. **Note** This parameter is supported only in some regions and will be supported in other regions in the future.
Path of Array	The path of the JSON array in the value of the Kafka message. This parameter is displayed only if you set the Output Mode parameter to Multi-row Output. If you want to reference the fields in a specific JSON object, you can specify a value for this parameter in the `a.a1` format. If you want to reference the fields in a specific JSON array, you can specify a value for this parameter in the `a[0].a1` format. If you leave this parameter empty, Kafka Reader parses the value of a message as a JSON array.

Parameter	Description
Configuration parameters	The extended parameters that you can configure when you create a Kafka producer. For example, you can configure the bootstrap.servers, auto.commit.interval.ms, and session.timeout.ms parameters. For more information about the parameters supported by Kafka clusters of different versions for Kafka producers, see Documentation of Apache Kafka. You can configure parameters in KafkaConfig to control the data write behavior of a Kafka producer. For a real-time synchronization node that synchronizes data from a Kafka data source, the default value of the acks parameter for a Kafka producer is all. If you require high performance for a Kafka producer, you can specify a different value for the acks parameter. Valid values of the acks parameter: ◦ 0: A Kafka producer does not acknowledge whether data is written to the destination. ◦ 1: A Kafka producer acknowledges that the write operation is successful if data is written to the primary replica. ◦ all: A Kafka producer acknowledges that the write operation is successful if data is written to all replicas.
Output Fields	The output fields, which can be customized. ◦ Click **Add more fields**. In the fields that appear, enter a **field name** and select a **data type** to customize a field. DataWorks provides two types of methods based on which Kafka Reader obtains values for fields from messages. You can click the ⇄ icon to switch between the two types of methods. ▪ Default methods: ▪ value: the values of messages ▪ key: the keys of messages ▪ partition: the IDs of partitions ▪ offset: the offsets of messages ▪ timestamp: the timestamps of messages, in milliseconds ▪ headers: the headers of messages

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>

- JSON-based parsing: You can use the .Sub-field or [Element in an array] syntax to obtain the content in the JSON format. To ensure that the values of fields are compatible with historical logic, you can use a string that starts with two underscores (_), such as __**value**__, to obtain specific values for fields from messages. The following code shows the data in a sample Kafka message:

<pre>
{
        "a": {
        "a1": "hello"
        },
        "b": "world",
        "c":[
                "xxxxxxx",
                "yyyyyyy"
                ],
        "d":[
                {
                        "AA":"this",
                        "BB":"is_data"
                },
                {
                        "AA":"that",
                        "BB":"is_also_data"
                }
           ]
}
</pre>

- You can use one of the following methods based on the preceding code:
  - If you want to read the values of messages, use __value__.
  - If you want to read the keys of messages, use __key__.
  - If you want to read the partitions that store messages, use __partition__.
  - If you want to read the offsets of messages, use __offset__.
  - If you want to read the timestamps of messages, use __timestamp__.
  - If you want to read the headers of messages, use __headers__.
  - If you want to read "hello" in the a1 field, use a.a1.
  - If you want to read "world" in the b field, use b.
  - If you want to read "yyyyyyy" in the c field, use c[1].
  - If you want to read "this" in the AA field, use d[0].AA.

- To remove a field, move the pointer over the field and click the 🗑 icon.

</td></tr>
</table>

7. Click the 💾 icon in the top toolbar.

> **Note** Kafka Reader in each synchronization node can read data from only one topic.

3.4.6.5. ApsaraDB for PolarDB reader

Currently, an ApsaraDB for PolarDB reader can only read data from ApsaraDB PolarDB for MySQL databases instead of ApsaraDB PolarDB for PostgreSQL databases.

Procedure

1. Go to the **DataStudio** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.

2. In the Scheduled Workflow pane, move the pointer over the +Create icon and choose **Data Integration > Real-time synchronization**.

   Alternatively, click the required workflow, right-click **Data Integration**, and then choose **Create > Real-time synchronization**.
3. In the **Create Node** dialog box, set the Sync Method parameter to **End-to-end ETL** and set the **Node Name** and **Location** parameters.

   > **Notice** The node name can be up to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).
4. Click **Commit**.
5. On the configuration tab of the real-time sync node, drag **PolarDB** under **Reader** to the editing panel.
6. Click **PolarDB1**. In the pane that appears on the right, set the required parameters in the **Node Settings** section.

Parameter	Description
Connection	The connection to the ApsaraDB PolarDB for MySQL data store. In this example, you can only select an ApsaraDB PolarDB for MySQL connection. If no connection is available, click **New data source** on the right to add one on the **Workspace Manage > Data Source** page.
Table	The name of the table from which data is read in the ApsaraDB PolarDB for MySQL data store. You can click **Preview** on the right to preview the selected table.
Output Fields	The fields from which data is read.

7. Click the 💾 icon in the top toolbar.

3.4.7. Writer

3.4.7.1. Configure MaxCompute Writer

MaxCompute provides a comprehensive data import scheme that supports the fast computing of large amounts of data.

Prerequisites

A reader or conversion node is configured. For more information, see Supported synchronization methods, sources, and destinations.

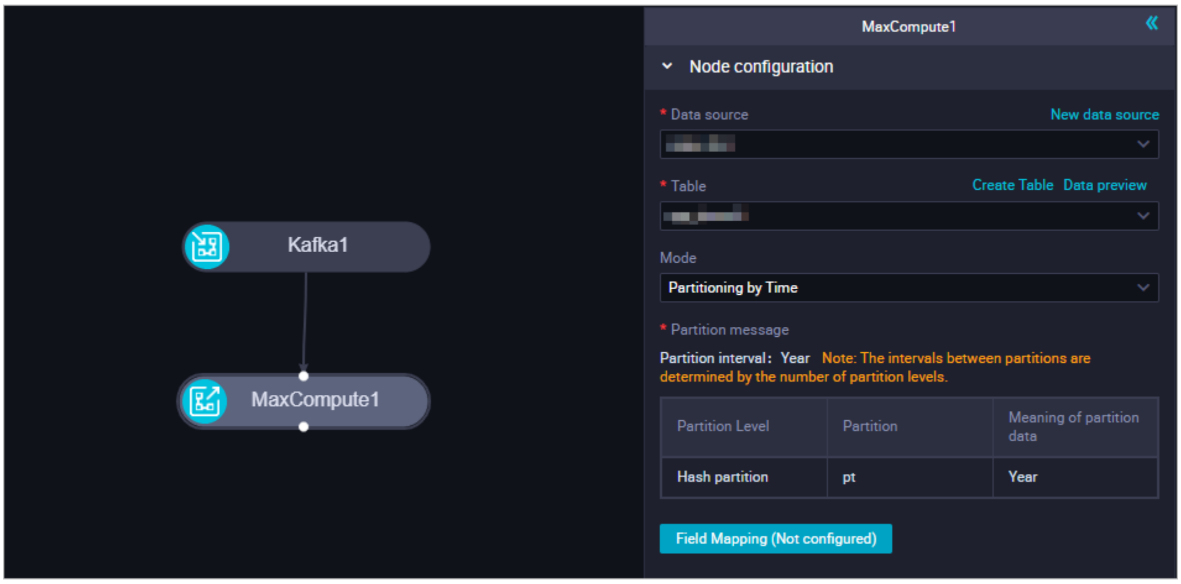
Procedure

1. Go to the **DataStudio** page.
   1. Log on to the DataWorks console.
   2. In the left-side navigation pane, click **Workspaces**.
   3. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
2. In the Scheduled Workflow pane, move the pointer over the +Create icon and choose **Data Integration > Real-time synchronization**.

   Alternatively, click the required workflow, right-click **Data Integration**, and then choose **Create > Real-time synchronization**.
3. In the **Create Node** dialog box, set the Sync Method parameter to **End-to-end ETL** and set the **Node Name** and **Location** parameters.

> **Notice** The node name can be up to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).

4. Click **Commit**.
5. On the configuration tab of the real-time sync node, drag **MaxCompute** in the **Output** section to the canvas on the right. Connect the MaxCompute node to the configured reader or conversion node.
6. Click the **MaxCompute** node. In the **Node configuration** panel, set the parameters. The following table describes the parameters.



MaxCompute1

Node configuration

* Data source New data source

[redacted]

* Table Create Table Data preview

[redacted]

Mode

Partitioning by Time

* Partition message

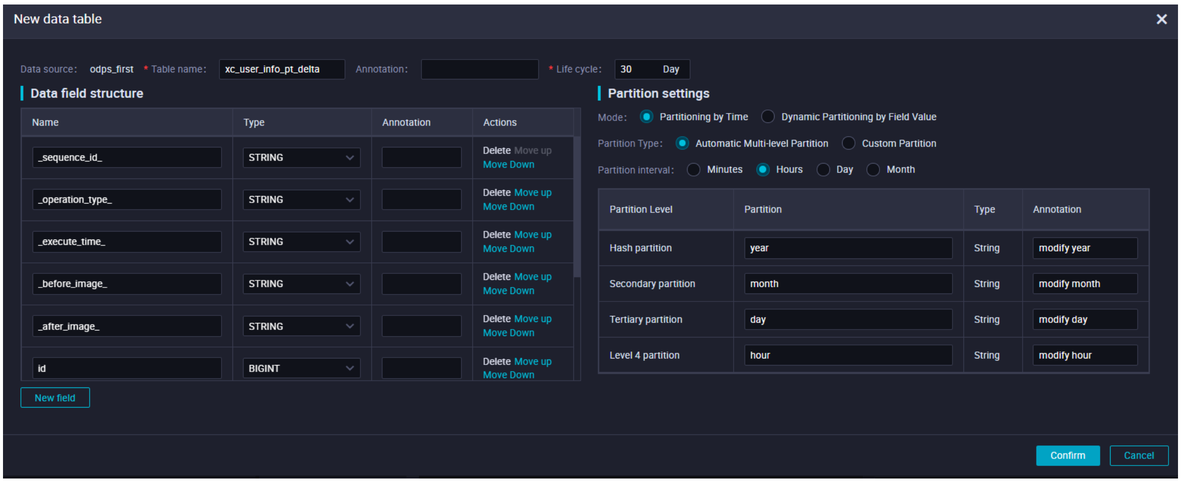
Partition interval: Year Note: The intervals between partitions are determined by the number of partition levels.

Partition Level	Partition	Meaning of partition data
Hash partition	pt	Year

Field Mapping (Not configured)

Parameter	Description
Data source	The MaxCompute data source that you configured. You can select only a MaxCompute data source. If no data source is available, click **New data source** on the right to add a data source on the **Data Source** page. For more information, see Add a MaxCompute data source.
Table	The name of the MaxCompute table to which you want to write data. You can click **Create Table** on the right to create a table, or click **Data preview** to preview the selected table. **Notice** Before you create a table, connect the MaxCompute node to a reader node and make sure that the output fields are specified for the reader node.
Mode	The mode in which data is written to the destination partitions of the MaxCompute table. Valid values: **Partitioning by Time** and **Dynamic Partitioning by Field Value**. If you select **Partitioning by Time**, data is written to the destination partitions of the MaxCompute table based on the value of the *_execute_time_* field. For more information, see Fields used for real-time synchronization. If you select **Dynamic Partitioning by Field Value**, data is dynamically written to the destination partitions of the MaxCompute table based on the value of a specified field in the source table after the mapping between the specified field in the source table and the specified partition field in the MaxCompute table is defined.
Partition message	The information about the partitioned MaxCompute table.
Field Mapping	The field mappings between the source and destination. Click **Field Mapping** to configure field mappings. The real-time sync node synchronizes data based on the field mappings.

If you want to create a table, click **Create Table** next to Table. In the **New data table** dialog box, set the parameters. The following table describes the parameters.



New data table

Data source: odps_first * Table name: xc_user_info_pt_delta Annotation: * Life cycle: 30 Day

Data field structure

Name	Type	Annotation	Actions
_sequence_id_	STRING		Delete Move up Move Down
_operation_type_	STRING		Delete Move up Move Down
_execute_time_	STRING		Delete Move up Move Down
_before_image_	STRING		Delete Move up Move Down
_after_image_	STRING		Delete Move up Move Down
id	BIGINT		Delete Move up Move Down

Partition settings

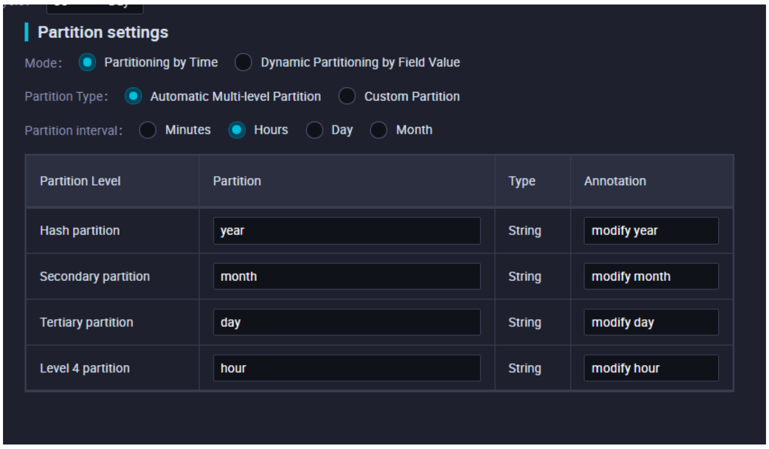
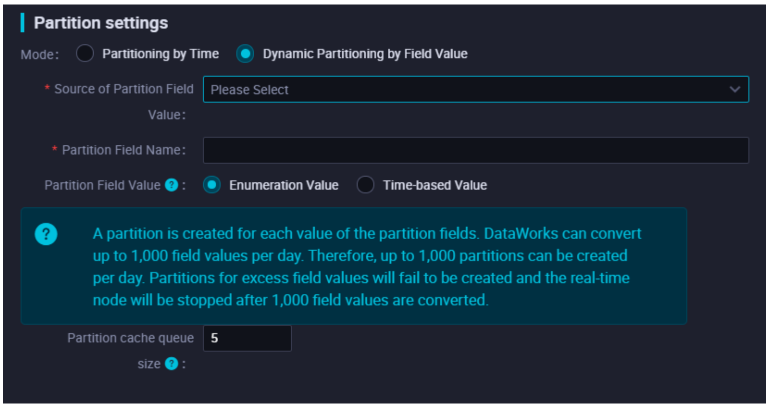
Mode: ● Partitioning by Time ○ Dynamic Partitioning by Field Value

Partition Type: ● Automatic Multi-level Partition ○ Custom Partition

Partition interval: ○ Minutes ● Hours ○ Day ○ Month

Partition Level	Partition	Type	Annotation
Hash partition	year	String	modify year
Secondary partition	month	String	modify month
Tertiary partition	day	String	modify day
Level 4 partition	hour	String	modify hour

Parameter	Description
Table name	The name of the MaxCompute table to which you want to write data in real time.
Life cycle	The lifecycle of the MaxCompute table. For more information, see Lifecycle.
Data field structure	The fields of the MaxCompute table. To add a field, click **New field**.

Parameter	Description
Partition settings	The partition information of the MaxCompute table. You can select **Partitioning by Time** or **Dynamic Partitioning by Field Value** as the partitioning mode. ○ **Partitioning by Time**: Data is written to the destination partitions of the MaxCompute table based on the value of the _execute_time_ field. For more information, see Fields used for real-time synchronization.  Partition settings Mode: ● Partitioning by Time ○ Dynamic Partitioning by Field Value Partition Type: ● Automatic Multi-level Partition ○ Custom Partition Partition interval: ○ Minutes ● Hours ○ Day ○ Month Partition Level | Partition | Type | Annotation Hash partition | year | String | modify year Secondary partition | month | String | modify month Tertiary partition | day | String | modify day Level 4 partition | hour | String | modify hour **Notice** ▪ You must configure at least two levels of partitions, which are yearly and monthly partitions. You can configure a maximum of five levels of partitions, which are yearly, monthly, daily, hourly, and minutely partitions. ▪ For more information about MaxCompute tables, see Partition. ○ **Dynamic Partitioning by Field Value**: Data is dynamically written to the destination partitions of the MaxCompute table based on the value of a specified field in the source table after the mapping between the specified field in the source table and the specified partition field in the MaxCompute table is defined.  Partition settings Mode: ○ Partitioning by Time ● Dynamic Partitioning by Field Value * Source of Partition Field Value: Please Select * Partition Field Name: Partition Field Value ?: ● Enumeration Value ○ Time-based Value A partition is created for each value of the partition fields. DataWorks can convert up to 1,000 field values per day. Therefore, up to 1,000 partitions can be created per day. Partitions for excess field values will fail to be created and the real-time node will be stopped after 1,000 field values are converted. Partition cache queue size ?: 5 For example, the value of Field A in the source table is defined as the value of the partition field in the MaxCompute table. If the value of Field A in a record is aa, this record is written to the aa partition of the MaxCompute table. If the value of Field A in a record is bb, this record is written to the bb partition of the MaxCompute table.

7. Click the 💾 icon in the toolbar.

3.4.7.2. Configure Hologres Writer

You can build a real-time data warehouse by using the real-time write capability of Hologres.

Prerequisites

A reader or conversion node is configured. For more information, see Plug-ins for data sources that support real-time synchronization.

Procedure

1. Go to the **DataStudio** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
2. In the Scheduled Workflow pane, move the pointer over the +Create icon and choose **Data Integration > Real-time synchronization**.

   Alternatively, click the required workflow, right-click **Data Integration**, and then choose **Create > Real-time synchronization**.
3. In the **Create Node** dialog box, set the Sync Method parameter to **End-to-end ETL** and set the **Node Name** and **Location** parameters.

   > **Notice** The node name can be up to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).
4. Click **Commit**.
5. On the configuration tab of the real-time synchronization node, drag **Hologres** in the **Output** section to the canvas on the right. Then, draw a line to connect it to the configured reader or conversion node.
6. Click the **Hologres** node. In the panel that appears, set the parameters as required.

Parameter	Description
Data source	The Hologres data source that you configured. You can select only a Hologres data source. If no data source is available, click **New data source** on the right to add a data source on the **Data Source** page. For more information, see Add a Hologres data source.
Table	The name of the Hologres table to which you want to write data. You can click **One-Click table creation** on the right to create a table, or click **Data preview** to preview the selected table.
Dynamic Time Partition	If the Hologres table is a partitioned table, you must specify a dynamic time-based partition. The dynamic time-based partition parses the value of a source field in the yyyymmdddhhmmss format. After the value is parsed, you can use the dynamic partition whose name is a string of variables in the destination table. The destination partition varies based on the value of the source field. For example, the value of the source field is 20200816, and the name of the destination partition is in the {yyyy}-{mm}-{dd} format. In this case, the value is written to the 2020-08-16 partition.

Parameter	Description
Job Type	The type of the data write operation. Valid values: **Replay (replay operation log to restore data)** and **Insert (direct archive save)**. ○ **Replay (replay operation log to restore data)**: Hologres Writer performs the same operation on the Hologres destination as that performed on the source. For example, if the `INSERT` statement is executed to add a record to the source, Hologres Writer executes the `INSERT` statement to add the same record to the Hologres destination. If the `UPDATE` or `DELETE` statement is executed in the source, Hologres Writer executes the `UPDATE` or `DELETE` statement in the Hologres destination. ○ **Insert (direct archive save)**: Hologres Writer uses the Hologres destination as streaming data storage. Data is synchronized from the source to the Hologres destination by using the `INSERT` statement.
Policy for Write Conflict	The solution to data write conflicts. Valid values: **Cover (Overwrite)** and **Ignore (Ignore)**. ○ **Cover (Overwrite)**: Hologres Writer uses the new data synchronized from the source to overwrite the existing data in the Hologres destination. ○ **Ignore (Ignore)**: Hologres Writer ignores the new data synchronized from the source and retains the existing data in the Hologres destination.
Field Mapping	The field mappings between the source and Hologres destination. Click **Field Mapping** to configure field mappings. The real-time synchronization node synchronizes data based on the field mappings.

7. Click the 💾 icon in the top toolbar.

3.4.7.3. Configure AnalyticDB for MySQL Writer

AnalyticDB for MySQL Writer allows you to build a real-time data warehouse by using the real-time write capability of AnalyticDB for MySQL.

Prerequisites

A reader or conversion node is configured. For more information, see Plug-ins for data sources that support real-time synchronization.

Procedure

1. Go to the **DataStudio** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. In the top navigation bar, select the region in which the workspace that you want to manage resides, find the workspace, and then click **Data Development** in the Actions column.
2. In the Scheduled Workflow pane of the DataStudio page, move the pointer over the +Create icon and choose **Data Integration > Real-time synchronization**.

   Alternatively, click the required workflow, right-click **Data Integration**, and then choose **Create > Real-time synchronization**.
3. Click **Commit**.
4. On the configuration tab of the real-time synchronization node, drag **AnalyticDB MySQL 3.0** in the **Output** section to the canvas on the right. Connect the AnalyticDB for MySQL node and the configured reader or conversion node.
5. Click the AnalyticDB for MySQL node. In the panel that appears, configure the parameters.

Parameter	Description
Data source	The AnalyticDB for MySQL data source that you have configured. You can select only an AnalyticDB for MySQL data source. If no data source is available, click **New data source** on the right to add a data source. For more information, see Configure an AnalyticDB for MySQL 3.0 connection.
Table	The name of the AnalyticDB for MySQL table to which you want to write data. You can click **One-Click table creation** on the right to create a table, or click **Data preview** to preview the selected table.
Write Mode	The write mode. The only value Replay is displayed. This value indicates that data is updated by row.
Field Mapping	The field mappings between the source and destination. Click **Field Mapping** to configure field mappings. The synchronization node synchronizes data based on the field mappings.

6. Click 💾 in the top toolbar to save the node.

3.4.7.4. Datahub writer

Datahub is a platform designed to process streaming data. You can publish and subscribe to applications for streaming data in Datahub and distribute the data to other platforms. Datahub allows you to analyze streaming data and build applications based on the streaming data.

The Datahub writer writes data to Datahub by using the Datahub SDK for Java. The SDK version is as follows:

```
<dependency>
    <groupId>com.aliyun.datahub</groupId>
    <artifactId>aliyun-sdk-datahub</artifactId>
    <version>2.5.1</version>
</dependency>
```

Create a Datahub writer

1. Log on to the DataWorks console. In the left-side navigation pane, click Workspaces. On the Workspaces page, find the target workspace and click **Data Analytics** in the Actions column.
2. On the Data Analytics tab, move the pointer over the +Create icon and choose **Data Integration** > **Real-Time Sync**.

   You can also find the target workflow, right-click **Data Integration**, and choose **Create** > **Real-Time Sync**.
3. In the **Create Node** dialog box that appears, set **Node Name** and **Location**, and then click **Commit**.
4. On the configuration tab of the created real-time sync node, drag **DataHub** under **Writer** to the editing panel. Connect it to the desired reader or transformation node in the panel.
5. Click the **Datahub writer** node and set parameters in the **Node Settings** section.

Parameter	Description
Connection	The connection to Datahub. In this example, you can only select a Datahub connection. If no connection is available, click **Add Connection** on the right to create one on the **Workspace Manage > Data Source** page.
Topic	The name of the topic to which data is written in Datahub. You can click **Preview** on the right to preview the selected topic.

Parameter	Description
Records per Batch	The number of records that are written at a time.
Mappings	The mappings between fields in the source and destination data stores. DataWorks synchronizes data based on the field mappings.

6. Click 💾 in the toolbar.

3.4.7.5. Configure Kafka Writer

To configure Kafka Writer, select a table and configure field mappings.

Prerequisites

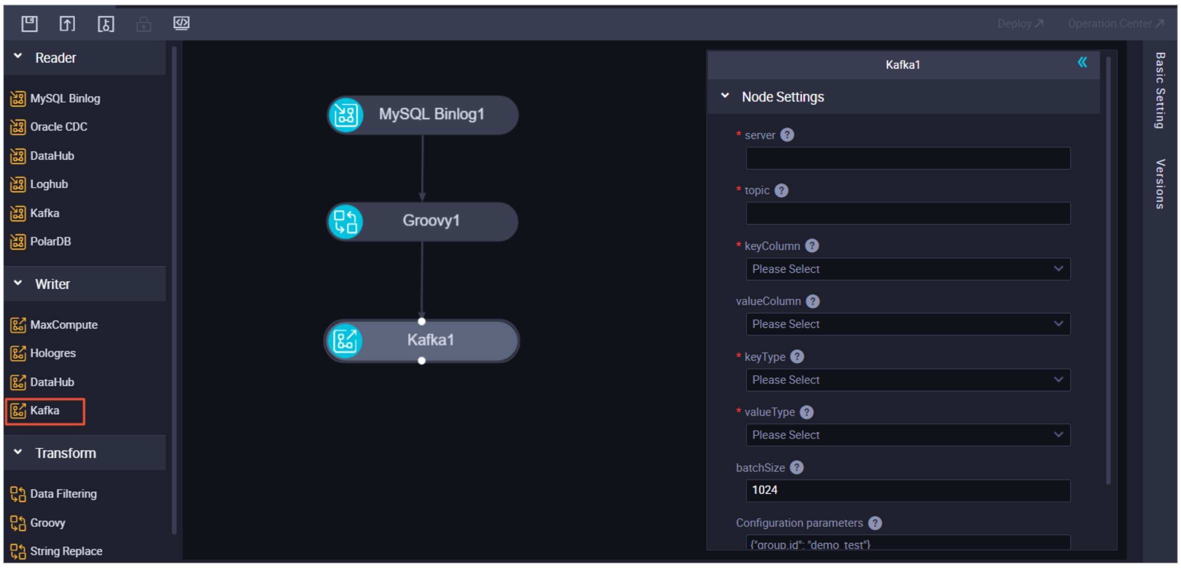
A reader or conversion node is configured. For more information, see Plug-ins for data sources that support real-time synchronization.

Procedure

1. Go to the **DataStudio** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
2. In the Scheduled Workflow pane, move the pointer over the +Create icon and choose **Data Integration > Real-time synchronization**.

   Alternatively, click the required workflow, right-click **Data Integration**, and then choose **Create > Real-time synchronization**.
3. In the **Create Node** dialog box, set the Sync Method parameter to **End-to-end ETL** and set the **Node Name** and **Location** parameters.

   > **Notice** The node name can be up to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).
4. Click **Commit**.
5. On the configuration tab of the real-time synchronization node, drag **Kafka** in the **Output** section to the canvas on the right. Connect the new node to the configured reader or conversion node.
6. Click the **Kafka** node. In the **Node Configuration** panel, set the parameters.



Kafka1

Node Settings

* server

* topic

* keyColumn: Please Select

valueColumn: Please Select

* keyType: Please Select

* valueType: Please Select

batchSize: 1024

Configuration parameters: {"group.id": "demo_test"}

Parameter	Description
Kafka Cluster Address	The address of the Kafka broker. Specify the address in the `IP address:Port number` format.
Topic	The name of the Kafka topic to which you want to write data. Kafka maintains feeds of messages in categories called topics. Each message that is published to a Kafka cluster is assigned to a topic. Each topic contains a group of messages.
Key Column	The column that is specified as the key.
Value Column	The column that is specified as the value. If you leave this parameter empty, all columns are concatenated by using the delimiter specified by the Column separator parameter to form the value.
Key Type	The data type of the keys in the Kafka topic.
Value Type	The data type of the values in the Kafka topic.
Number of Bytes Written at a Time	The number of data records to write at a time. Default value: *1024*.
Configuration parameters	The extended parameters specified when KafkaConsumer is created, such as the bootstrap.servers, auto.commit.interval.ms, and session.timeout.ms parameters. You can set parameters in kafkaConfig to control the data consumption behavior of KafkaConsumer. For a real-time synchronization node that synchronizes data to a Kafka data source, the default value of the acks parameter for KafkaProducer is all. If you have higher requirements for performance, you can specify a different value for the acks parameter. Valid values of the acks parameter: ○ 0: does not check data writes. ○ 1: checks whether data is written to a topic and its replicas as expected. ○ all: checks whether data is written to all replicas of a topic as expected.

7. Click in the toolbar.

3.4.7.6. Configure Elasticsearch Writer

You can build a real-time data warehouse by using the real-time write capability of Elasticsearch.

Prerequisites

A reader or conversion node is configured. For more information about the data sources that support real-time synchronization, see Plug-ins for data sources that support real-time synchronization.

Limits

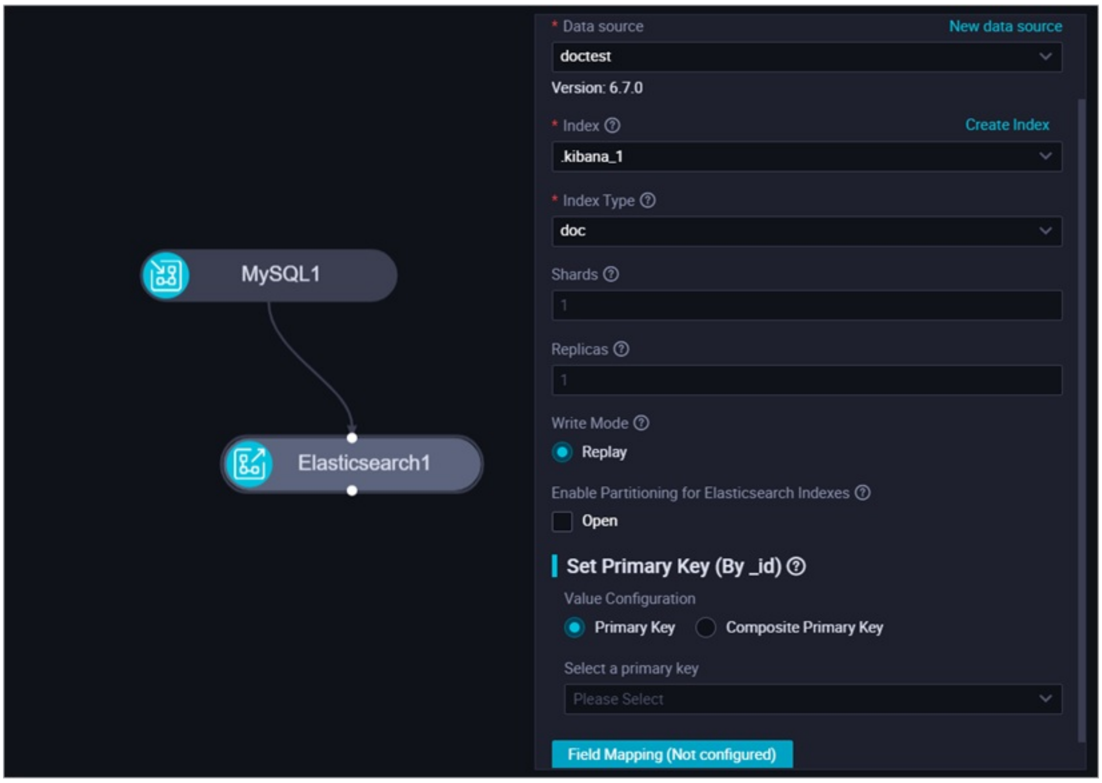
DataWorks allows you to add Alibaba Cloud Elasticsearch V5.X, V6.X, and V7.X clusters as data sources. Self-managed Elasticsearch clusters are not supported.

Procedure

1. Go to the **DataStudio** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
2. In the Scheduled Workflow pane, move the pointer over the +Create icon and choose **Data Integration > Real-time synchronization**.

   Alternatively, click the required workflow, right-click **Data Integration**, and then choose **Create > Real-time synchronization**.
3. In the **Create Node** dialog box, set the Sync Method parameter to **End-to-end ETL** and set the **Node Name** and **Location** parameters.

   > **Notice** The node name can be up to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).
4. Click **Commit**.
5. On the configuration tab of the real-time synchronization node, drag **Elasticsearch** in the **Output** section to the canvas. Connect it to the configured reader or conversion node in the canvas.
6. Click the **Elasticsearch** node. In the panel that appears, configure the parameters.



* Data source: doctest

Version: 6.7.0

* Index: .kibana_1

* Index Type: doc

Shards: 1

Replicas: 1

Write Mode: ● Replay

Enable Partitioning for Elasticsearch Indexes: □ Open

Set Primary Key (By _id)

Value Configuration: ● Primary Key ○ Composite Primary Key

Select a primary key: Please Select

Field Mapping (Not configured)

Parameter	Description
Data source	The Elasticsearch data source that you configured. In this example, you can select only an Elasticsearch data source. If no data source is available, click **New data source** on the right to add a data source on the **Data Source** page. For more information, see Add an Elasticsearch data source.
Index	The name of the index to which you want to write data. You can click Create Index on the right to create an index. You can directly use the default index information to create an index. Alternatively, you can modify the settings of Index Name, Index Type, Dynamic Mapping Status, Shards, Replicas, and Statement for Creating Index and create an index. ○ **Index Type**: This parameter is available only for Elasticsearch V6.X, V5.X, or earlier. ○ **Dynamic Mapping Status**: This parameter is used to specify the value of the dynamic parameter. The dynamic parameter determines whether Elasticsearch Writer dynamically writes new fields to the mappings of the index. ■ If you use an Elasticsearch cluster whose version is earlier than V7.10, this parameter has the following valid values: true, false, and strict. ■ If you use an Elasticsearch cluster whose version is V7.10 or later, this parameter has the following valid values: true, false, strict, and runtime. where: ■ true: indicates that Elasticsearch Writer writes new fields to the mappings of the index and the fields can be searched. ■ false: indicates that Elasticsearch Writer writes new fields to the mappings of the index but the fields cannot be searched. ■ strict: indicates that if Elasticsearch Writer detects new fields, it returns an error message and does not write the fields to the mappings of the index. ■ runtime: indicates that Elasticsearch Writer writes new fields to the mappings of the index as runtime fields but the fields cannot be searched. For more information, see the dynamic parameter for open source Elasticsearch. ○ **Shards**: the number of primary shards. An index can be divided into multiple primary shards. These primary shards can be distributed among different nodes to support distributed searches. When you create an index, you must specify the number of primary shards for the index. After the index is created, you cannot change the number. For more information, see shard. ○ **Replicas**: the number of replica shards for each primary shard. The replica shards can be used for fault tolerance and to process the read request workloads of the cluster. If the capacity of the cluster is insufficient, only a single backup is required for each primary shard, or the cluster encounters bottlenecks in write performance, set Replicas to 1. ○ **Statement for Creating Index**: The field configurations are configured in properties. You can modify the types of the fields.
Enable Partitioning for Elasticsearch Indexes	Specifies whether to enable the routing mechanism. You can customize the value of the routing parameter. The default value of routing is the ID of a document. A Hash function is used to convert the value of routing to obtain a number. The number is used to divide the number of primary shards to obtain a remainder. The remainder indicates the position of the document in the primary shards.
Set Primary Key (By_Id)	Set the method used to assign values to the IDs of Elasticsearch indexes during data synchronization. ○ Primary Key: uses one of the columns in the source table as the primary key. ○ Composite Primary Key: combines multiple columns in the source table to form the primary key.

Parameter	Description
Field Mapping	Configure field mappings between the source and destination. The synchronization node synchronizes data based on the field mappings.

7. Click the 💾 icon in the toolbar.

3.4.8. Transformation

3.4.8.1. Configure Data Filtering

The Data Filtering plug-in can filter data based on specified rules, such as the field size. Only data that meets the rules is retained.

Prerequisites

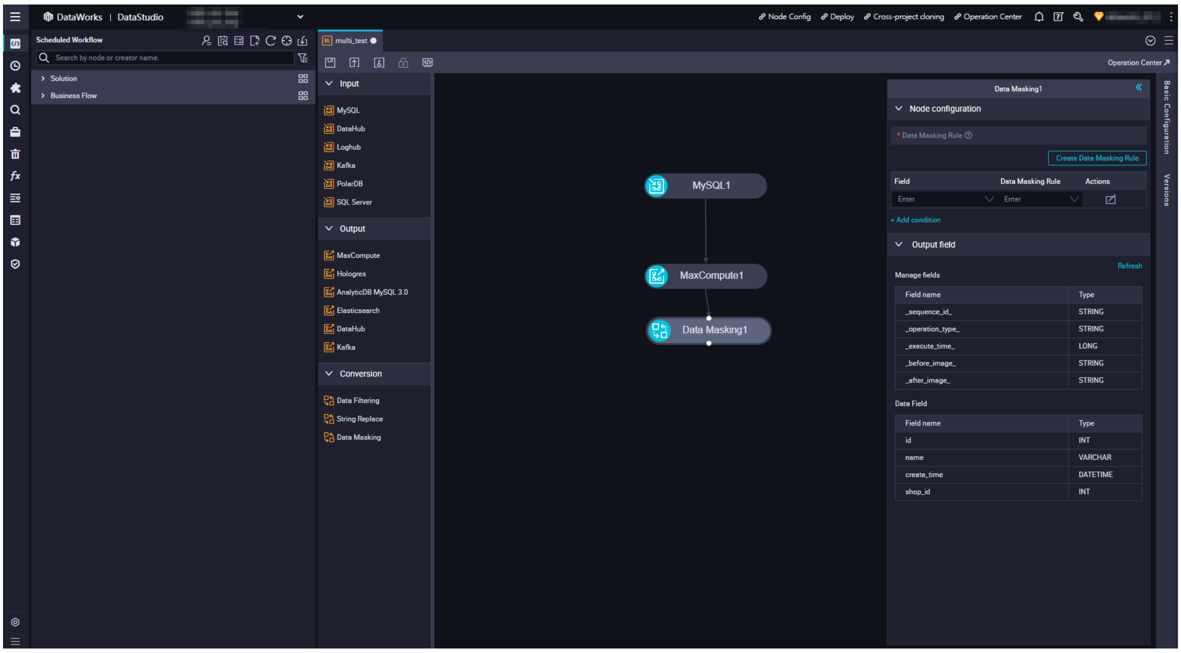
A reader node is configured. For more information, see Plug-ins for data sources that support real-time synchronization.

Procedure

1. Go to the **DataStudio** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
2. In the Scheduled Workflow pane, move the pointer over the +Create icon and choose **Data Integration > Real-time synchronization**.

   Alternatively, click the required workflow, right-click **Data Integration**, and then choose **Create > Real-time synchronization**.
3. In the **Create Node** dialog box, set the Sync Method parameter to **End-to-end ETL** and set the **Node Name** and **Location** parameters.

   > **Notice** The node name can be up to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).
4. Click **Commit**.
5. On the configuration tab of the real-time sync node, drag **Data Filtering** under **Conversion** to the canvas on the right. Connect the new node to a reader node.
6. Click the new **Data Filtering** node. In the configuration pane that appears, set the required parameters in the **Node configuration** section.



Data Masking1

Node configuration

* Data Masking Rule

Create Data Masking Rule

Field	Data Masking Rule	Actions
Enter	Enter	

+ Add condition

Output field

Refresh

Manage fields

Field name	Type
_sequence_id_	STRING
_operation_type_	STRING
_execute_time_	LONG
_before_image_	STRING
_after_image_	STRING

Data Field

Field name	Type
id	INT
name	VARCHAR
create_time	DATETIME
shop_id	INT

- **Node configuration**

  Rules: the rules for filtering data in data stores. Only data that meets the rules is retained.

- **Output field**

  The names and types of output fields after filtering.

7.

3.4.8.2. String replacement

String replacement is a transformation plug-in used to replace field values of the STRING type.

1. Log on to the DataWorks console. In the left-side navigation pane, click Workspaces. On the Workspaces page, find the target workspace and click **Data Analytics** in the Actions column.
2. On the Data Analytics tab, move the pointer over the +Create icon and choose **Data Integration** > **Real-Time Sync**.

   You can also find the target workflow, right-click **Data Integration**, and choose **Create** > **Real-Time Sync**.
3. In the **Create Node** dialog box that appears, set **Node Name** and **Location**, and then click **Commit**.
4. On the configuration tab of the created real-time sync node, drag **String Replace** under **Transform** to the editing panel. Connect it to the desired reader in the panel.
5. Click the **string replacement** node and set parameters in the **Node Settings** section.
   - **Node Settings**

     Rule:

     - **Field**: the field of the parent node to be used as the input field.
     - **Regular Expression Match**: specifies whether a regular expression is used to search for the original string.
     - **Original String**: the original string to search.
     - **New String**: the new string to replace the original string.
     - **Case Sensitive**: specifies whether the value is case sensitive during the search.

     Add Condition: Click the button to add more string replacement rules.
   - **Output Fields**

     The output fields after string replacement.
6. Click 💾 in the toolbar.

3.4.8.3. Configure data de-identification

The data de-identification feature can de-identify sensitive data in a single table that is synchronized in real time and store the de-identified data to a specified database.

Prerequisites

A reader is configured. For more information, see Plug-ins for data sources that support real-time synchronization.

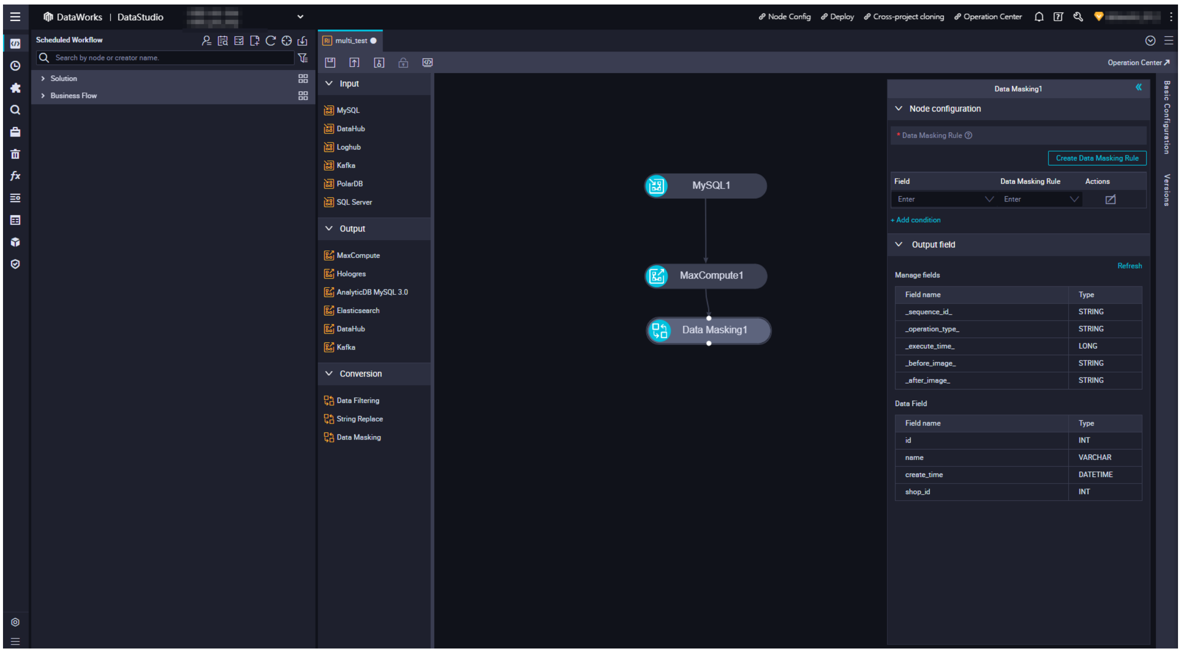
Procedure

1. Go to the **DataStudio** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
2. In the Scheduled Workflow pane, move the pointer over the **+Create** icon and choose **Data Integration > Real-time synchronization**.

   Alternatively, click the required workflow, right-click **Data Integration**, and then choose **Create > Real-time synchronization**.
3. In the **Create Node** dialog box, set the Sync Method parameter to **End-to-end ETL** and set the **Node Name** and **Location** parameters.

   > **Notice** The node name can be up to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).

4. Click **Commit**.
5. On the configuration tab of the real-time sync node, drag **Data Masking** under **Conversion** to the canvas on the right.
6. Click the **Data Masking** node. In the configuration panel that appears, set the parameters.



Data Masking1

Node configuration

* Data Masking Rule

Create Data Masking Rule

Field	Data Masking Rule	Actions
Enter	Enter	

+ Add condition

Output field

Refresh

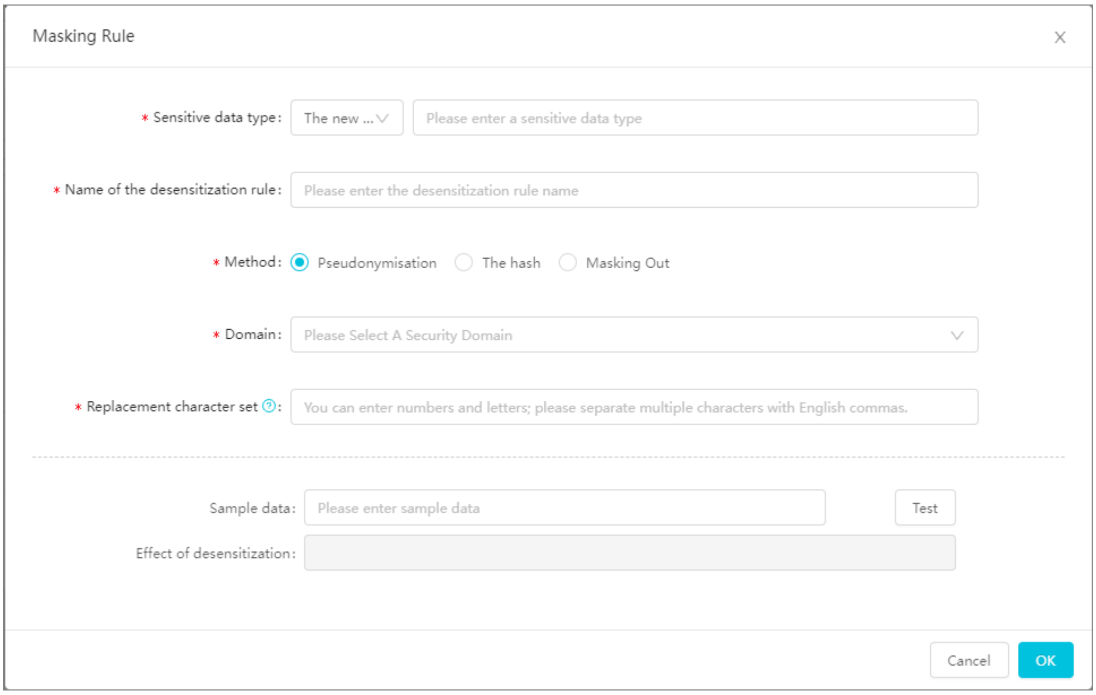
Manage fields

Field name	Type
_sequence_id_	STRING
_operation_type_	STRING
_execute_time_	LONG
_before_image_	STRING
_after_image_	STRING

Data Field

Field name	Type
id	INT
name	VARCHAR
create_time	DATETIME
shop_id	INT

   i. Create a data de-identification rule. Click **Create Data Masking Rule**. In the **Create Data Masking Rule** dialog box that appears, set the **Sensitive Data Type**, **Rule Name**, **Data Masking Method**, **Security Domain**, and **Character Set for Replacement** parameters.
      a. Create a data de-identification rule.



Masking Rule

* Sensitive data type: The new ... | Please enter a sensitive data type

* Name of the desensitization rule: Please enter the desensitization rule name

* Method: ● Pseudonymisation ○ The hash ○ Masking Out

* Domain: Please Select A Security Domain

* Replacement character set: You can enter numbers and letters; please separate multiple characters with English commas.

Sample data: Please enter sample data

Effect of desensitization:

a. Set basic parameters.

Parameter	Description
Sensitive Data Type	■ By default, **Existing Data Type** is selected from the drop-down list on the left. You can select an existing sensitive data type from the drop-down list on the right. The existing sensitive data types include built-in sensitive data types and sensitive data types created by all users. ■ You can also select **New Data Type** from the drop-down list on the left. In the field on the right, enter the name of the sensitive data type. The name must be 1 to 30 characters in length and can contain letters and digits. After you enter the name of a new data type, the system checks whether the name is used by existing sensitive data types, including built-in sensitive data types and sensitive data types created by all users. If the name has been used, the message **The specified sensitive field type already exists.** is displayed. **Note** The built-in sensitive data types are Mobile Phone Number, Id Card, Bank Card, Email, IP, Car No, Post Code, Seat Number, Mac Address, Address, Name, Company, Nation, Constellation, Gender, and Nationality.
Rule Name	By default, the system populates this field with the value of the **Sensitive Data Type** parameter. You can change the rule name. The name must be 1 to 30 characters in length and can contain letters and digits. If you enter a name that has been used by an existing rule, the message **The specified rule name already exists.** is displayed.

b. Configure the data de-identification method. DataWorks supports the following data de-identification methods: **pseudonymisation**, **hashing**, and **redaction**.

- **Pseudonymisation**

  This method replaces the characters of a data record with an artificial pseudonym of the same data type. The format of the pseudonym is the same as that of the original data record.

  - If you set the **Sensitive Data Type** parameter to a built-in sensitive data type, such as Mobile Phone Number, Id Card, Bank Card, Email, IP, Car No, Post Code, Seat Number, Mac Address, Address, Name, or Company, you must set the **Security Domain** parameter for your data records.

    Security Domain: You can select a digit from 0 to 9 from the Security Domain drop-down list. Data de-identification policies vary with the security domain. Different data de-identification results are returned for the same data record in different security domains. For example, if the data record is a123 and the security domain is set to 0, the data de-identification result is b124. If the security domain is set to 1, the data de-identification result is c234. In a security domain, the same data de-identification result is returned for a data record at all times.

  - If you set the **Sensitive Data Type** parameter to a custom sensitive data type, you must set the **Character Set for Replacement** parameter for your data records.

    Character Set for Replacement: Enter the characters to be replaced. Separate multiple characters with commas (,). The characters can be letters or digits. If a data record contains a character that is specified in this field, the character is replaced with another character of the same type. For example, if a data record contains digits from 0 to 3 and letters from a to d, the data de-identification result also contains only characters within these ranges. If a character is not included in this field, the character is not replaced.

- **Hashing**

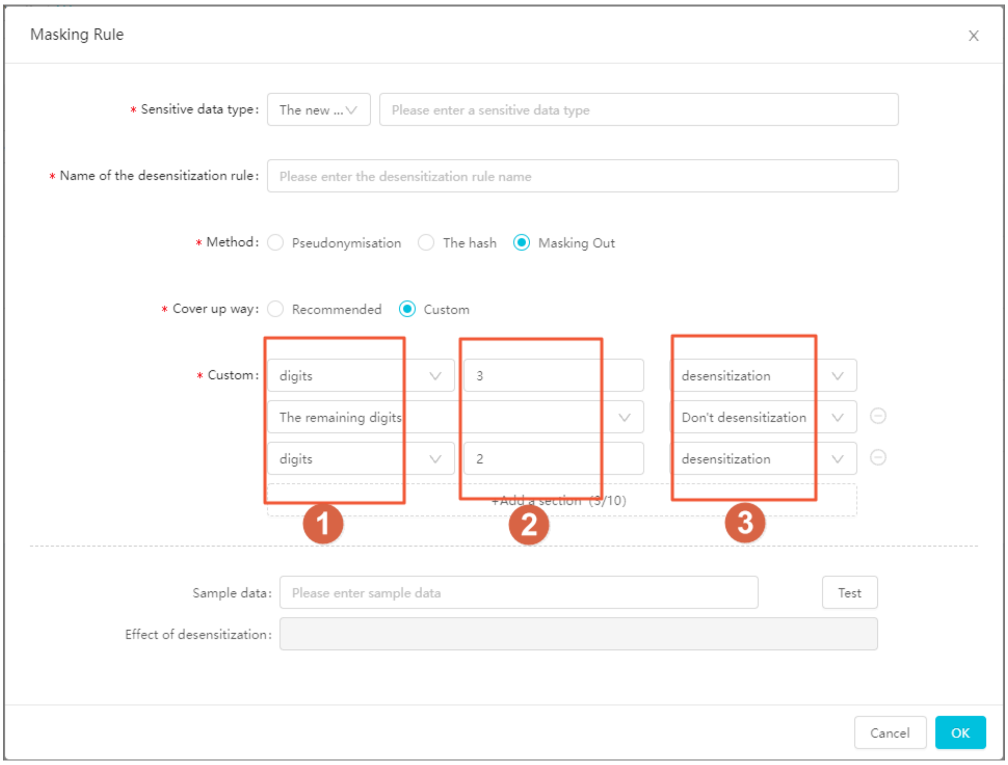
  This method encrypts a data record to generate a hash value of a fixed length. If you select this method, you must set the **Security Domain** parameter.

  Security Domain: You can select a digit from 0 to 9 from the Security Domain drop-down list. Data de-identification policies vary with the security domain. Different data de-identification results are returned for the same data record in different security domains. For example, if the data record is a123 and the security domain is set to 0, the data de-identification result is b124. If the security domain is set to 1, the data de-identification result is c234. In a security domain, the same data de-identification result is returned for a data record at all times.

- **Redaction**

  This method replaces each of the characters at specific positions of a data record with an asterisk (*).

  - **Recommendation Method**: If you select Recommendation Method for the Redaction Mode parameter, select **Only show first and last character**, **Show first three and last two characters**, or **Show first three and last four characters** from the Recommendation Method drop-down list. By default, Only show first and last character is selected.
  - **Custom**: You can flexibly specify whether to de-identify the specified number of characters at the first, middle, or last part of a data record. You can add up to 10 segments. You must add at least one **Remaining Digits** segment.



Masking Rule

* Sensitive data type: The new ... | Please enter a sensitive data type

* Name of the desensitization rule: Please enter the desensitization rule name

* Method: ○ Pseudonymisation ○ The hash ● Masking Out

* Cover up way: ○ Recommended ● Custom

* Custom:

digits	3	desensitization
The remaining digits		Don't desensitization
digits	2	desensitization

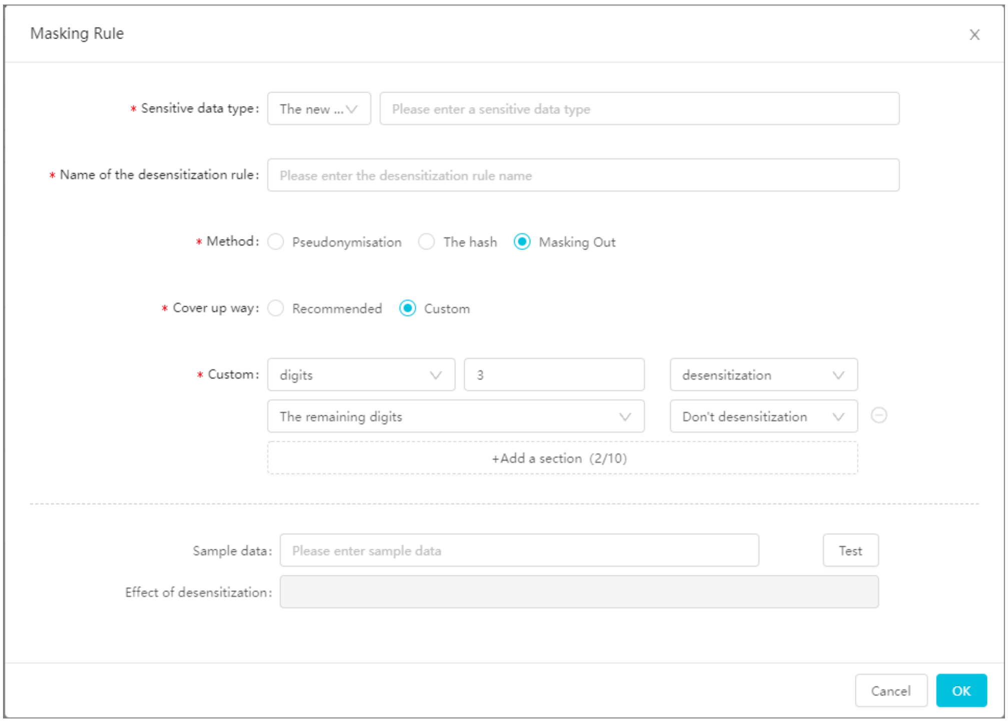
+Add a section (3/10)

Sample data: Please enter sample data

Effect of desensitization:

Icon	Description
①	You can select **Digits** or **Remaining Digits**.
②	You can enter an integer from 1 to 100.
③	You can select **Mask** or **Do Not Mask**.

The following figure shows how to de-identify the first three digits and leave the remaining digits intact.



Masking Rule

* Sensitive data type: The new ... | Please enter a sensitive data type

* Name of the desensitization rule: Please enter the desensitization rule name

* Method: ○ Pseudonymisation ○ The hash ● Masking Out

* Cover up way: ○ Recommended ● Custom

* Custom:

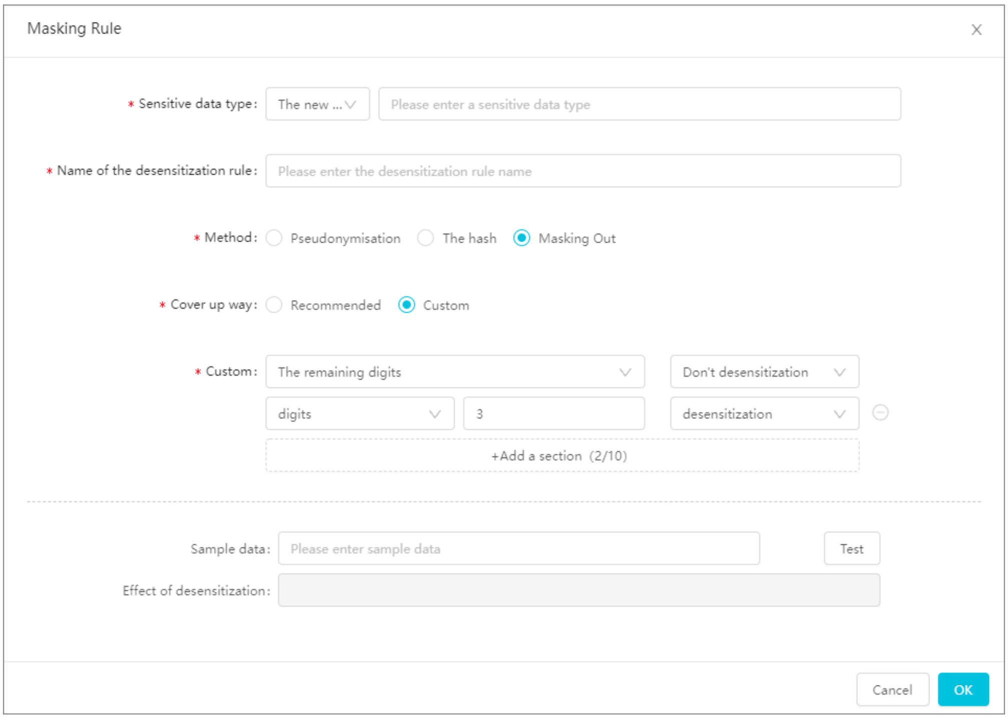
digits	3	desensitization
The remaining digits		Don't desensitization

+Add a section (2/10)

Sample data: Please enter sample data

Effect of desensitization:

The following figure shows how to de-identify the last three digits and leave the remaining digits intact.



Masking Rule

* Sensitive data type: The new ... | Please enter a sensitive data type

* Name of the desensitization rule: Please enter the desensitization rule name

* Method: ○ Pseudonymisation ○ The hash ● Masking Out

* Cover up way: ○ Recommended ● Custom

* Custom: The remaining digits | Don't desensitization

digits | 3 | desensitization

+Add a section (2/10)

Sample data: Please enter sample data | Test

Effect of desensitization:

c. Verify the data de-identification rule. You can enter sample data in the **Sample Data** field. The sample data can be 0 to 100 characters in length. Click **Test**. The data de-identification result is displayed in the **Data Masking Effect** field.

b. Click **OK**. In the configuration panel, you can select this newly added rule from the Data Masking Rule drop-down list for a field to be de-identified. The newly created rule is also synchronized to Data Security Guard.

ii. Click **Add condition** to add a row. In this row, you can configure the data de-identification rule for another field.

- In the **Field** column, select an output field of the parent node of the data de-identification node from the drop-down list.
- In the **Data Masking Rule** column, select a rule from the drop-down list. The rules that can be selected are those that have taken effect in Data Security Guard.
- In the **Actions** column of a field, click the **Edit** icon.
  - If the data de-identification rule for this field is created by you, you can modify the rule in the **Edit Data Masking Rule** dialog box that appears. You can enter sample data to verify the rule.
  - If the data de-identification rule is not created by you, you can check the configuration details of the rule in the dialog box. You can also enter sample data to verify the rule.
- In the **Actions** column of a field, click the **Delete** icon to delete the field.

iii. In the **Output field** section, select the fields to be used as output fields from the fields of the original table.

3.5. Synchronize all data in a database to MaxCompute

3.5.1. Plan and configure resources

When you use DataWorks to synchronize data, you can use only exclusive resource groups for data integration to run real-time data synchronization nodes. This topic describes the resources and configurations required to run real-time data synchronization nodes.

Context

- Resource planning and preparation

Before you use a data synchronization node to synchronize data, you must purchase an exclusive resource group for data integration and add the resource group to DataWorks for subsequent use.

For more information about exclusive resource groups for Data Integration, see Exclusive resources for Data Integration.

- Network connections

  An exclusive resource group for Data Integration is essentially a group of Elastic Compute Service (ECS) instances. After you purchase and create such an exclusive resource group, it is isolated from other services. You must associate the resource group with a virtual private cloud (VPC) to ensure network connectivity between the resource group and data sources during subsequent data synchronization.

What's next

After you plan and configure resources, you can configure data sources. You must configure network connectivity for the data sources and permissions to access the data sources. This facilitates the creation of a real-time data synchronization node. For more information about how to configure data sources, see Configure a data source (PolarDB), Configure data sources for data synchronization from Oracle, and Configure data sources for data synchronization from MySQL.

3.5.2. Configure a data source (PolarDB)

If you want to synchronize data from PolarDB to MaxCompute in real time, the source is PolarDB, and the destination is MaxCompute. Before you run a data synchronization task, you must refer to the operations in this topic to prepare the configurations, such as network environments and whitelists, for both the source and destination.

Prerequisites

Before you configure a data source, make sure that the following operations are performed:

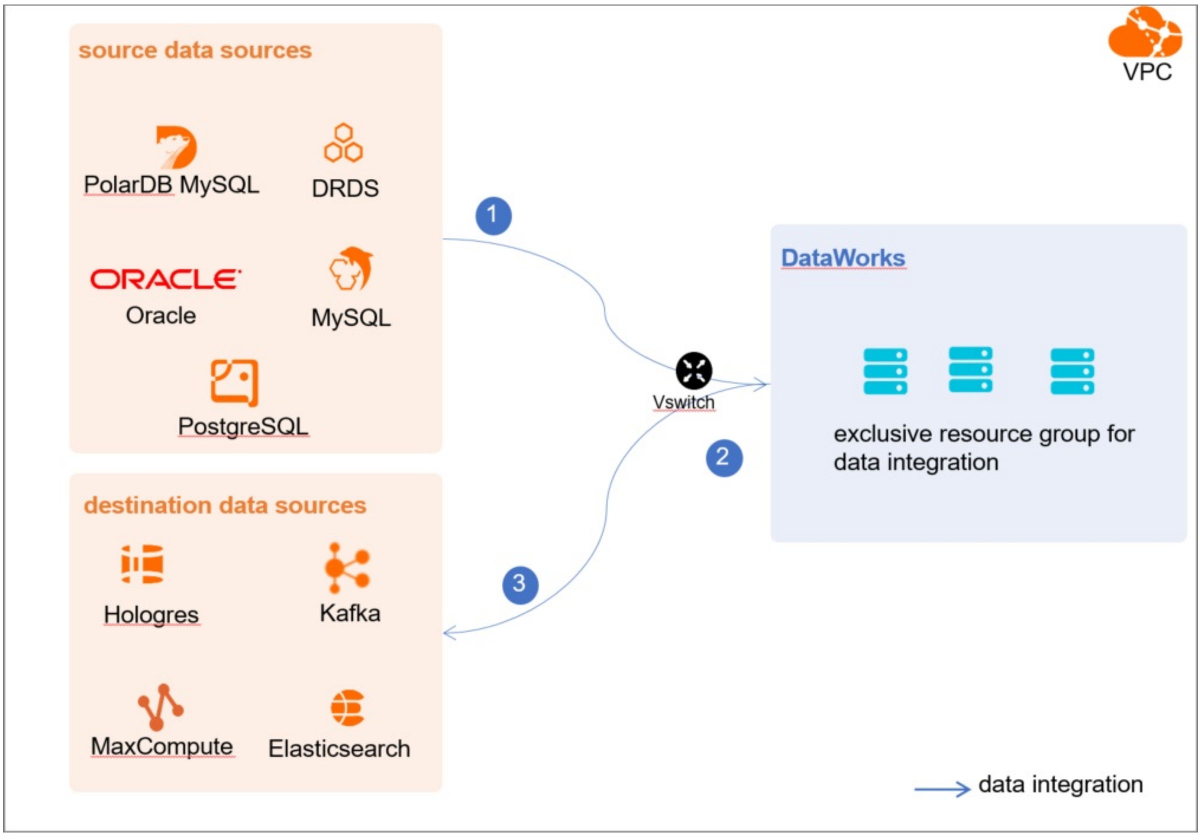
- Prepare data sources: A PolarDB for MySQL cluster and a MaxCompute project are created. In this topic, a PolarDB for MySQL cluster is used as the source data source.
- Plan and prepare resources: An exclusive resource group for data integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, connect data sources to exclusive resource groups for data integration based on your business requirements. After data sources and exclusive resource groups for data integration are connected, you can refer to the operations in this topic to configure access settings such as vSwitches and whitelists.
  - If data sources and exclusive resource groups for data integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If data sources and exclusive resource groups for data integration reside in different network environments, you must connect data sources and resource groups by using methods such as a VPN gateway.
- Prepare the MaxCompute client: The MaxCompute client is installed. You need to use the MaxCompute client to configure attributes for the destination MaxCompute data source. For more information, see MaxCompute client.

Context

Before you synchronize data from source data sources to destination data sources, make sure that data sources and exclusive resource groups are connected. In addition, you must make sure that exclusive resource groups can be used to access data sources.

- Configure whitelists for data sources

If data sources and exclusive resource groups for data integration reside in the same VPC, you need to add the CIDR block of the exclusive resource group to the whitelists of data sources. This ensures that the exclusive resource group for data integration can be used to access data sources.



- Create an account and authorize the account

  You must create an account that can be used to access data sources, read data from the source data source, and write data to the destination data source in the data integration process.

- Enable the binary logging feature

  If the source data source is a PolarDB for MySQL cluster, you must enable the binary logging feature for the cluster. Alibaba Cloud PolarDB for MySQL is fully compatible with MySQL and uses high-level physical logs to replace binary logs. To facilitate the integration between PolarDB and the MySQL ecosystem, you can enable the binary logging feature for PolarDB clusters.

Limits

- Only PolarDB for MySQL clusters can be used as source data sources. In this topic, PolarDB indicates PolarDB for MySQL data sources.
- Only data stored on the primary node of the PolarDB cluster can be synchronized.

Configure the source PolarDB data source

1. Configure a whitelist for the PolarDB for MySQL cluster.

   To add the CIDR block of the VPC where the exclusive resource group for Data Integration resides to a whitelist of the PolarDB for MySQL cluster, perform the following steps:

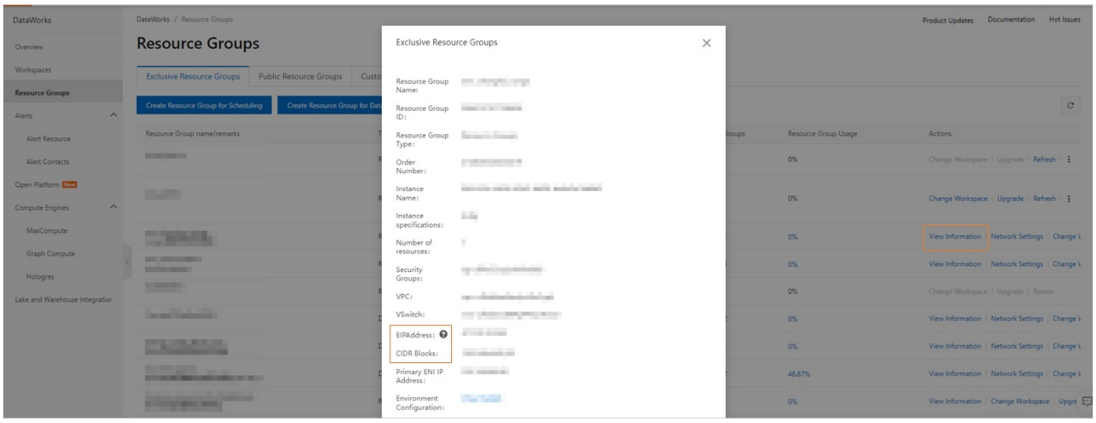
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

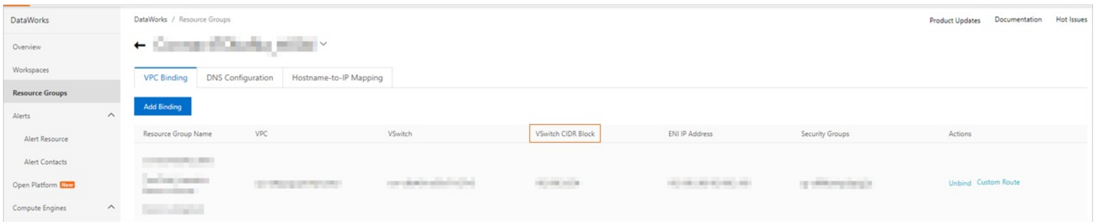
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.

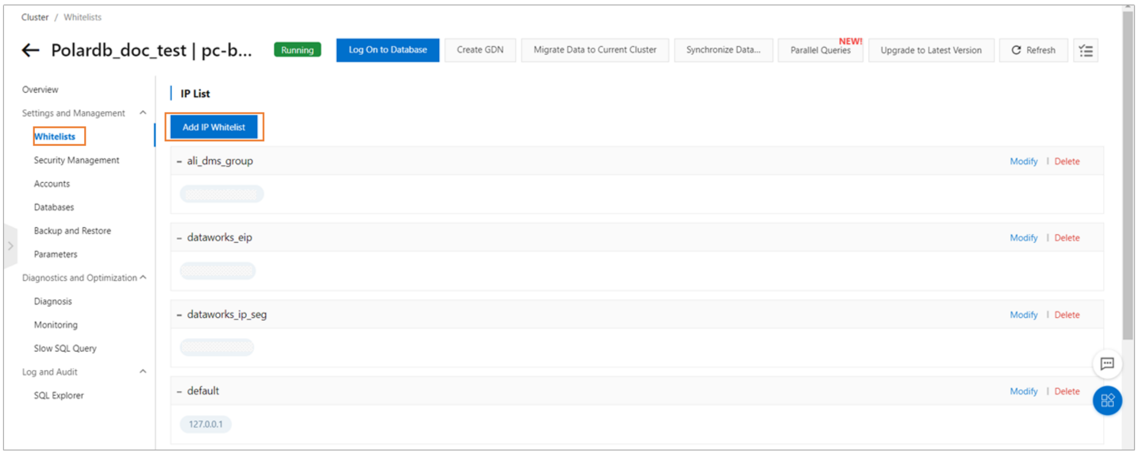


e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR blocks recorded in the preceding steps to the whitelist of the PolarDB for MySQL cluster.



For more information, see Configure an IP whitelist.

2. Create a PolarDB for MySQL database account.

For more information, see Create a database account.

3. Enable the binary logging feature for the PolarDB cluster.

For more information, see Enable binary logging.

Configure the destination MaxCompute data source

1. Log on to the MaxCompute client by using the account of a project owner.

   For more information, see MaxCompute client.

2. Enable the atomicity, consistency, isolation, durability (ACID) property for the MaxCompute project.

   Run the following command on the MaxCompute client:

   ```
   setproject odps.sql.acid.table.enable=true;
   ```

3. (Optional)Enable the MaxCompute V2.0 data type edition.

   If you need to use the TIMESTAMP data type in MaxCompute V2.0, run the following command to enable the MaxCompute V2.0 data type edition:

   ```
   setproject odps.sql.type.system.odps2=true;
   ```

4. Create an Alibaba Cloud account.

   This account is used to add a data source and access MaxCompute for data synchronization. For more information about how to create an Alibaba Cloud account, see Create an Alibaba Cloud account.

   After the Alibaba Cloud account is created, you can record the AccessKey ID and AccessKey secret of the account for future use.

What's next

After data sources are configured, the source data source, destination data source, and exclusive resource group for data integration are connected. Then, the exclusive resource group for data integration can be used to access data sources. You can add the source data source and destination data source to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

3.5.3. Configure data sources for data synchronization from Oracle

Before you synchronize data from Oracle to MaxCompute in real time, you can refer to the operations described in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions. You must configure a source Oracle data source and a destination MaxCompute data source.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

- Prepare data sources: An Oracle database and a MaxCompute project are prepared.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.
- Prepare the MaxCompute client: The MaxCompute client is installed. You need to use the MaxCompute client to configure attributes for the destination MaxCompute data source. For more information, see MaxCompute client.

Context

Configure a source Oracle data source

Configure the destination MaxCompute data source

1. Log on to the MaxCompute client by using the account of a project owner.

For more information, see MaxCompute client.

2. Enable the atomicity, consistency, isolation, durability (ACID) property for the MaxCompute project.

   Run the following command on the MaxCompute client:

   ```
   setproject odps.sql.acid.table.enable=true;
   ```

3. (Optional)Enable the MaxCompute V2.0 data type edition.

   If you need to use the TIMESTAMP data type in MaxCompute V2.0, run the following command to enable the MaxCompute V2.0 data type edition:

   ```
   setproject odps.sql.type.system.odps2=true;
   ```

4. Create an Alibaba Cloud account.

   This account is used to add a data source and access MaxCompute for data synchronization. For more information about how to create an Alibaba Cloud account, see Create an Alibaba Cloud account.

   After the Alibaba Cloud account is created, you can record the AccessKey ID and AccessKey secret of the account for future use.

What's next

After data sources are configured, the source data source, destination data source, and exclusive resource group for data integration are connected. Then, the exclusive resource group for data integration can be used to access data sources. You can add the source data source and destination data source to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

3.5.4. Configure data sources for data synchronization from MySQL

Before you synchronize data from a MySQL data source to MaxCompute in real time, you can refer to the operations in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions.

Prerequisites

Before you configure the data sources, make sure that the following operations are performed:

- Prepare data sources: A MySQL data source and a destination MaxCompute data source are created.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.
- Execute the following statement to check the version of your MySQL database. You can use a `MySQL 5.x` or `MySQL 8.x` database.

  ```
  select version();
  ```

  Note Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.x` or `V8.x`. PolarDB for MySQL is not supported. If the MySQL version of your ApsaraDB RDS for MySQL database is not `V5.x` or `V8.x`, use an ApsaraDB RDS for MySQL database whose MySQL version is `V5.x` or `V8.x`. Otherwise, the data synchronization node fails to run.

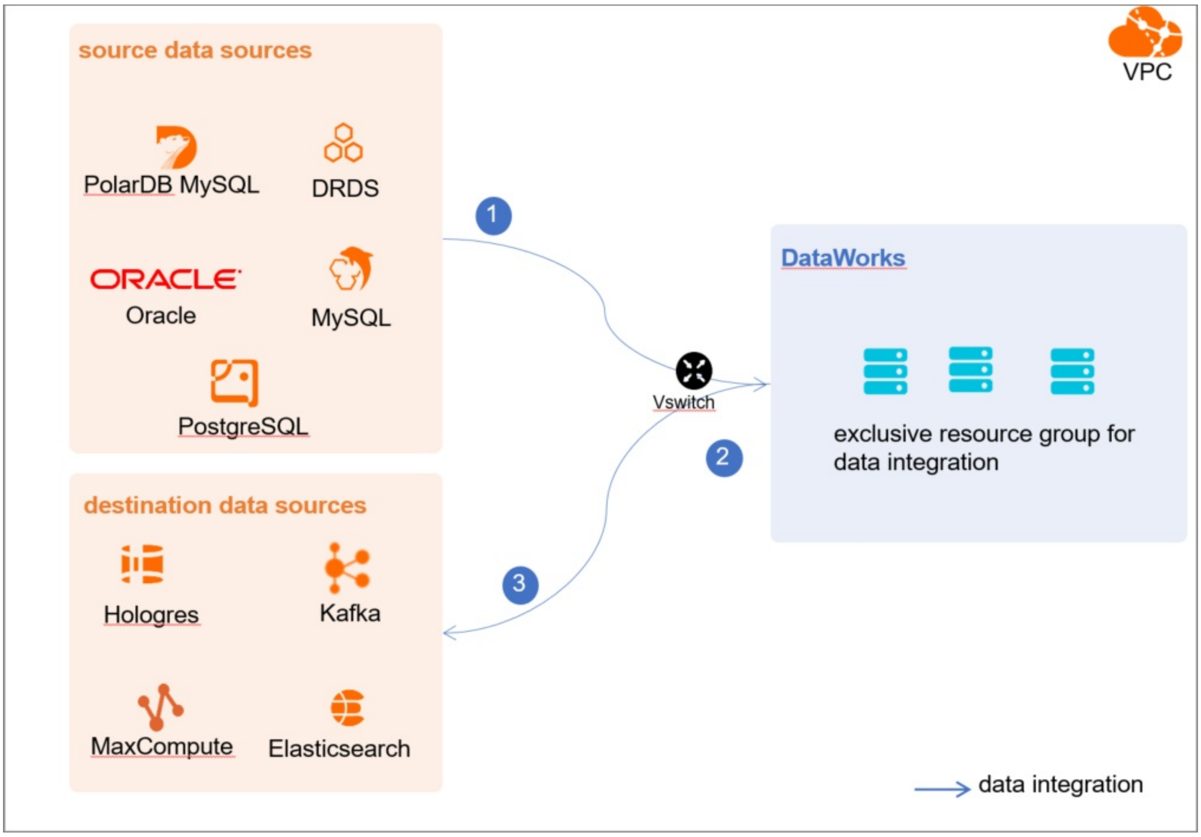
- Prepare the MaxCompute client: The MaxCompute client is installed. You need to use the MaxCompute client to configure attributes for the destination MaxCompute data source. For more information, see MaxCompute client.

Context

Before you synchronize data from source data sources to destination data sources, make sure that data sources and exclusive resource groups for data integration are connected. You must also make sure that the exclusive resource groups for data integration can be used to access the data sources.

- Configure whitelists for the data sources

  If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source data source is a MySQL database, you must enable the binary logging feature. Binary logs record changes to all table schemas and modifications on table data. You can execute statements such as CREATE and ALTER to perform operations on table schemas. You can execute statements such as INSERT, UPDATE, and DELETE to perform operations on table data. You can use binary logs to view the change history of the database, back up incremental data and restore data in the database, and replicate data from the primary database to secondary databases.

  Formats of binary logs:

  - Statement: SQL statement-based replication. Binary logs in this format record the SQL statements that are executed to modify data entries.
  - Row: row-based replication. Binary logs in this format record only modification details about data entries in rows.

- Mixed: replication in mixed mode. This mode combines the statement and row formats. In most cases, binary logs in the statement format are used to record the SQL statements that are executed to modify data entries, such as functions. If data replication from the primary database to secondary databases cannot be implemented by using binary logs in this format, switch to the row format. MySQL determines which format to use based on each SQL statement that is executed.

Limits

Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.X` or `V8.X`. PolarDB for MySQL is not supported.

Configure a source MySQL data source

1. Configure a whitelist for the MySQL database.

   Add the CIDR block of the VPC in which the exclusive resource group resides to the whitelist of the MySQL database.

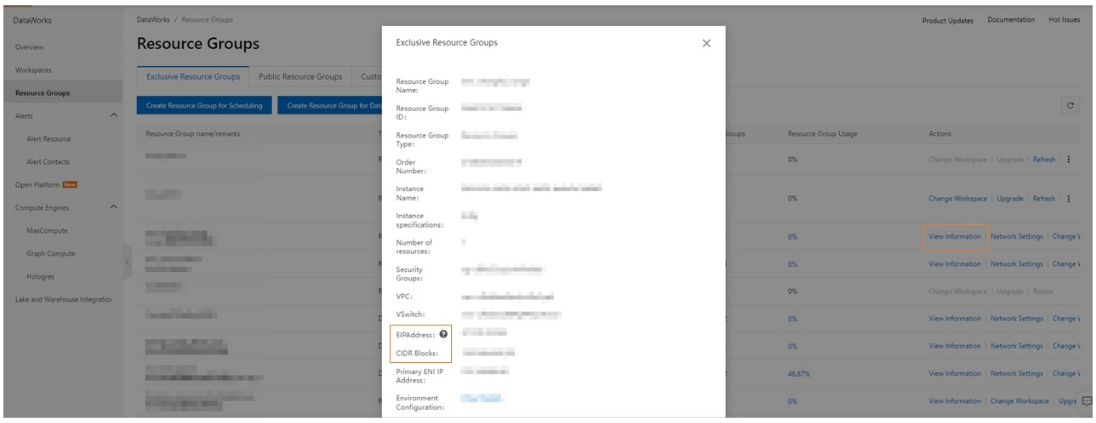
   i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

      a. Log on to the DataWorks console.

      b. In the left-side navigation pane, click **Resource Groups**.

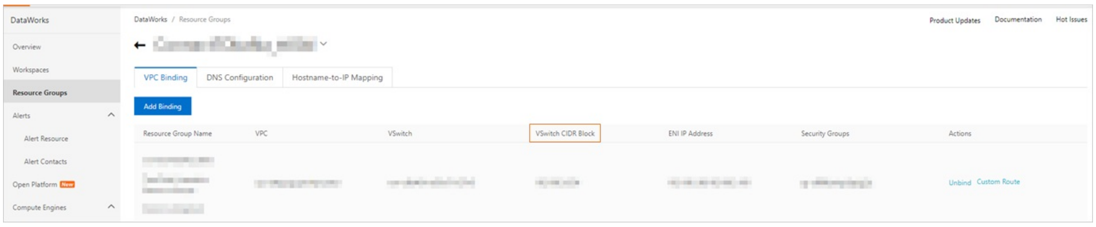
      c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

      d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



      e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

      f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



   ii. Add the EIP and CIDR block of the exclusive resource group to the whitelist of the MySQL database.

2. Create an account and grant permissions to the account.

   You must create an account to log on to the MySQL database. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

   i. Create an account.

      For more information, see Create an account to access a MySQL database.

ii. Grant permissions to the account.

You can run the following command to grant permissions to the account. Alternatively, you can grant the `SUPER` permission to the account. Replace `Account for data synchronization` with the created account when you execute the specific statement.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Password'; // Create an account that is used to synchronize data and set a password so that you can use the account and password to access the database from a host. % indicates a host.
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%'; // Grant the SELECT, REPLICATION SLAVE, and REPLICATION CLIENT permissions to the account.
```

`*.*` indicates that the synchronization account is granted the preceding permissions on all tables in all databases. You can also grant the preceding permissions on specified tables in the specified database to the synchronization account. For example, to grant the account the preceding permissions on the *user* table in the *test* database, execute the following statement: `GRANT SELECT, REPLICATION CLIENT ON test.user TO 'Account for data synchronization'@'%';`.

> **Note** The `REPLICATION SLAVE` permission is a global permission. You cannot grant this permission on specified tables in the specified database to the synchronization account.

3. Enable the binary logging feature for the MySQL database.

Perform the following steps to check whether the binary logging feature is enabled and to query the format of binary logs:

- Execute the following statement to check whether the binary logging feature is enabled:

```
show variables like "log_bin";
```

If *ON* is displayed in the returned result, the binary logging feature is enabled.

- If you use a secondary database to synchronize data, execute the following statement to check whether the binary logging feature is enabled:

```
show variables like "log_slave_updates";
```

If *ON* is displayed in the returned result, the binary logging feature is enabled for the secondary database.

If the returned result is different from the preceding result, follow the instructions described in the *MySQL documentation* to enable the binary logging feature.

Execute the following statement to view the format of binary logs:

```
show variables like "binlog_format";
```

Returned result:

- *ROW*: The format of binary logs is row.
- *STATEMENT*: The format of binary logs is statement.
- *MIXED*: The format of binary logs is mixed.

Configure the destination MaxCompute data source

1. Log on to the MaxCompute client by using the account of a project owner.

For more information, see MaxCompute client.

2. Enable the atomicity, consistency, isolation, durability (ACID) property for the MaxCompute project.

Run the following command on the MaxCompute client:

```
setproject odps.sql.acid.table.enable=true;
```

3. (Optional)Enable the MaxCompute V2.0 data type edition.

If you need to use the TIMESTAMP data type in MaxCompute V2.0, run the following command to enable the MaxCompute V2.0 data type edition:

```
setproject odps.sql.type.system.odps2=true;
```

4. Create an Alibaba Cloud account.

   This account is used to add a data source and access MaxCompute for data synchronization. For more information about how to create an Alibaba Cloud account, see Create an Alibaba Cloud account.

   After the Alibaba Cloud account is created, you can record the AccessKey ID and AccessKey secret of the account for future use.

What's next

After data sources are configured, the source data source, destination data source, and exclusive resource group for data integration are connected. Then, the exclusive resource group for data integration can be used to access data sources. You can add the source data source and destination data source to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

3.5.5. Add a data source

Before you configure a real-time data synchronization node to synchronize data from a data source to the MaxCompute data source, you must add both data sources to DataWorks for subsequent source and destination configurations.

Prerequisites

Add a source Oracle data source

To add a source Oracle data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source Oracle data source, see Add an Oracle data source.

Add a source MySQL data source

To add a source MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source MySQL data source, see Add a MySQL data source.

Add a destination MaxCompute data source

For more information, see Add a MaxCompute data source.

What's next

For more information, see Configure and manage a real-time data synchronization node.

3.5.6. Configure and manage a real-time data sync node

After you prepare data sources, network environments, and resources, you can create a real-time data syn node to synchronize data to MaxCompute. This topic describes how to create a real-time sync node and view the status of the node.

Prerequisites

Before you create a real-time data sync node, make sure that the following operations are performed:

- Plan and configure resources
- Configure a data source (PolarDB)
- Configure data sources for data synchronization from Oracle
- Configure data sources for data synchronization from MySQL
- Add a data source

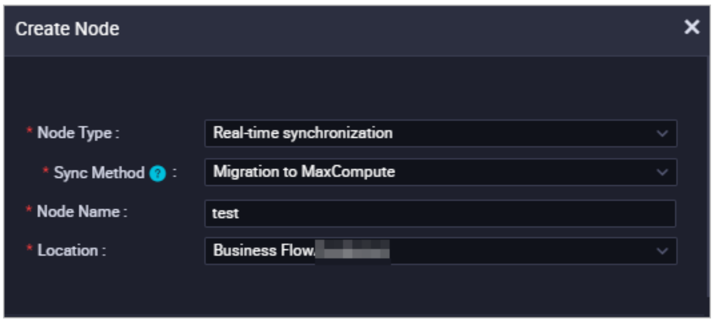
Limits

- DataWorks allows you to use only exclusive resource groups for Data Integration to run real-time data sync nodes.
- You can run real-time data sync nodes to synchronize data only from PolarDB, Oracle, or MySQL to MaxCompute.
- A real-time data sync node cannot be used to synchronize data in a table that has no primary key.

Create a real-time data sync node

1. Log on to the DataWorks console.
2. In the left-side navigation pane, click **Workspaces**.
3. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
4.
5. Create a real-time data sync node.
    1. On the DataStudio page, move the pointer over the +Create icon and choose **Data Integration > Real-time synchronization**.

       Alternatively, find the workflow in which you want to create a real-time data sync node and right-click the **Data Integration**. From the shortcut menu, choose **Create > Real-time synchronization**.
    2. In the **Create Node** dialog box, set the parameters that are described in the following table.



Create Node

* Node Type : Real-time synchronization

* Sync Method : Migration to MaxCompute

* Node Name : test

* Location : Business Flow/[redacted]

Parameter	Description
Node Type	The type of the node. Default value: **Real-time synchronization**.
Sync Method	Set the value to **Migration to MaxCompute**. In this case, partial or all tables in your database are migrated to MaxCompute.
Node Name	The name of the node. The name must be 1 to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).
Location	The directory in which the real-time data syn node is stored.

    3. Click **Commit**. You are navigated to the configuration tab of the real-time data sync node.
6. Select a resource group.
    1. On the right side of the configuration tab, click the **Basic Configuration** tab.
    2. In the panel that appears, select the resource group that you want to use from the **Resource Group** drop-down list.

> **Note**
>
> DataWorks allows you to use only exclusive resource groups for Data Integration to run real-time data sync nodes.
>
> If no exclusive resource group for Data Integration exists, click **Create Exclusive Resource Group for Data Integration** to create a resource group. For more information, see Overview.

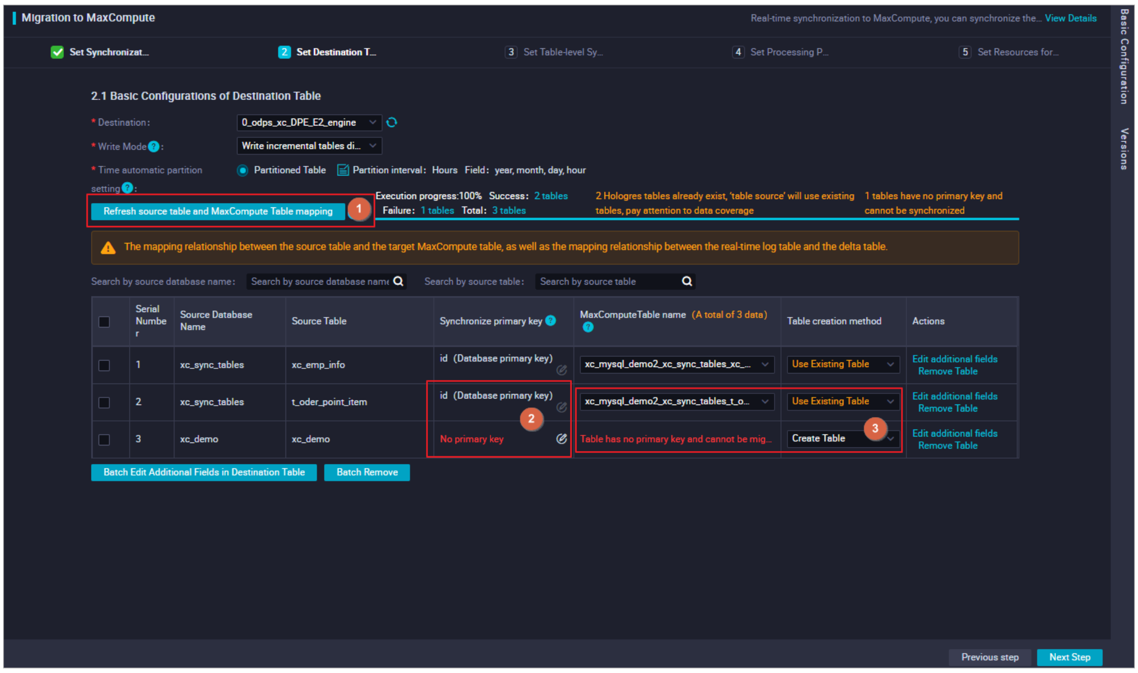
7.
8. Select a data source as the destination and configure the formats for the destination tables.
    1. In the **Set Destination Table** step, configure the **Destination** and **Write Mode** parameters.
    2. Click the icon next to **Automatic Partitioning by Time**. In the **Edit** dialog box, modify the partition settings for the destination tables. You can configure daily partitions. You can write data to a partitioned table or a non-partitioned table in MaxCompute.

iii. (Optional)Add fields to the destination tables

If you want to add fields to all the tables to be synchronized, click **New field** in the **Fields In Destination Table** section.

iv. Click **Refresh source table and MaxCompute Table mapping** to create mappings between the source tables and destination MaxCompute tables.

v. View the mapping progress, source tables, and mapped destination tables.



2.1 Basic Configurations of Destination Table

* Destination: 0_odps_xc_DPE_E2_engine

* Write Mode: Write incremental tables di...

* Time automatic partition setting: ● Partitioned Table Partition interval: Hours Field: year, month, day, hour

Refresh source table and MaxCompute Table mapping

Execution progress:100% Success: 2 tables Failure: 1 tables Total: 3 tables

2 Hologres tables already exist, 'table source' will use existing tables, pay attention to data coverage

1 tables have no primary key and cannot be synchronized

The mapping relationship between the source table and the target MaxCompute table, as well as the mapping relationship between the real-time log table and the delta table.

Search by source database name: Search by source database name Search by source table: Search by source table

□	Serial Number	Source Database Name	Source Table	Synchronize primary key	MaxComputeTable name (A total of 3 data)	Table creation method	Actions
□	1	xc_sync_tables	xc_emp_info	id (Database primary key)	xc_mysql_demo2_xc_sync_tables_xc_...	Use Existing Table	Edit additional fields Remove Table
□	2	xc_sync_tables	t_oder_point_item	id (Database primary key)	xc_mysql_demo2_xc_sync_tables_t_o...	Use Existing Table	Edit additional fields Remove Table
□	3	xc_demo	xc_demo	No primary key	Table has no primary key and cannot be mig...	Create Table	Edit additional fields Remove Table

Batch Edit Additional Fields in Destination Table   Batch Remove

Area No.	Description
1	The progress of mapping the source tables to the destination tables. Note The mapping may require a long period of time if data is synchronized from a large number of tables.
2	■ If the tables in the source database contain primary keys, the system removes duplicate data based on the primary keys during the synchronization. ■ If the tables in the source database do not contain primary keys, you can click the ✎ icon to customize primary keys. You can use one field or a combination of several fields as the primary keys of the tables. This way, the system removes duplicate data based on the primary keys during the synchronization. Note A real-time data sync node cannot be used to synchronize data in a table that has no primary key.

Area No.	Description
3	The name of the destination table. The table name that appears in the MaxComputeBase Table name column varies based on the method that you select from the drop-down list in the **Table creation method** column. ■ If you select **Create Table** from the drop-down list in the Table creation method column, the name of the destination table that is automatically created appears. You can click the table name to view and modify the table creation statement. ■ If you select **Use Existing Table** from the drop-down list in the Table creation method column, you must select a table name from the drop-down list in the MaxComputeBase Table name column. **Note** If a source table does not contain the primary key, you can click the edit icon next to No primary key in the Synchronized Primary Key column and specify the primary key for the source table so that full and incremental data can be synchronized from the source table.

vi. Click **Next Step**.

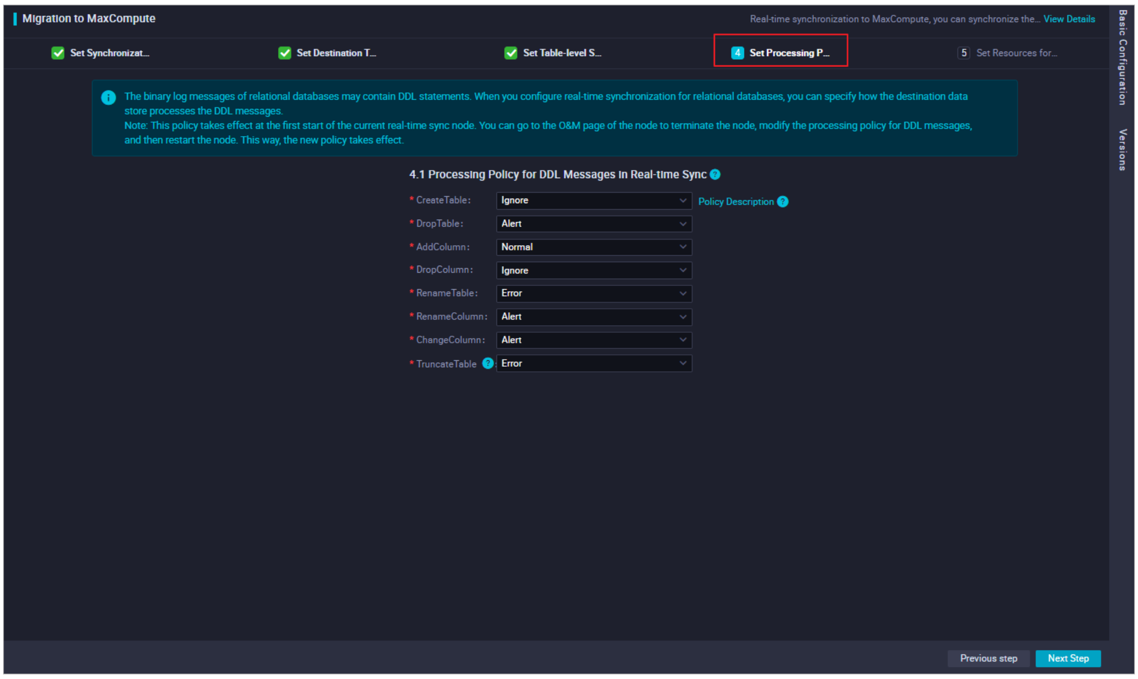
If you set the **Table creation method** to **Create Table**, you must click **Start table building** in the **Create tables automatically** dialog box to create destination MaxCompute tables.

9. Configure rules for processing DDL messages.

DDL statements exist in the source. Before you synchronize data, you can configure synchronization rules for different DDL statements based on your business requirements.

Note The rules apply when a real-time data sync node is run for the first time. If you want to modify the rules in subsequent operations, go to the Real Time DI page of Operation Center. For more information, see Manage the real-time data sync node.

i. In the **Set Processing Policy for DDL Messages** step, configure rules to process DDL messages during data synchronization.



The binary log messages of relational databases may contain DDL statements. When you configure real-time synchronization for relational databases, you can specify how the destination data store processes the DDL messages.
Note: This policy takes effect at the first start of the current real-time sync node. You can go to the O&M page of the node to terminate the node, modify the processing policy for DDL messages, and then restart the node. This way, the new policy takes effect.

4.1 Processing Policy for DDL Messages in Real-time Sync

- * CreateTable: Ignore
- * DropTable: Alert
- * AddColumn: Normal
- * DropColumn: Ignore
- * RenameTable: Error
- * RenameColumn: Alert
- * ChangeColumn: Alert
- * TruncateTable: Error

The following table describes the processing rules for different DDL messages.

DDL message	Rule
CreateTable	DataWorks processes a DDL message of the related type based on the following rules after it receives the message: ■ **Normal**: sends the message to the destination. Then, the destination processes the message. Each destination may process DDL messages based on its own business logic. If you select Normal for CreateTable, DataWorks only forwards the messages. ■ **Ignore**: ignores the message and does not send it to the destination. ■ **Alert**: ignores the message and records the alert in real-time synchronization logs. In addition, the alert contains information about the reason indicating that a message is ignored because of a running error. ■ **Error**: returns an error when the real-time sync solution is running and terminates the real-time sync solution.
DropTable	
AddColumn	
DropColumn	
RenameTable	
RenameColumn	
ChangeColumn	
TruncateTable	

ii. Click **Next**.

10. Configure the resources required by the data sync node.

i. In the **Set Resources for Solution Running** step, set the parameters that are described in the following table.

Parameter	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: **15**.
Maximum number of parallel threads allowed to read by destination	The maximum number of parallel threads that the sync node uses to read data from the source table or write data to the destination. Maximum value: 32. Specify an appropriate number based on the resources of the source and the destination.

ii. Click **Complete Configuration**.

Commit and deploy the real-time data sync node

1.

Start the real-time data sync node

1. Go to the Operation Center page.

   After you commit and deploy the real-time data sync node, click **Operation Center** in the upper-right corner of the DataStudio page to manage the node on the **Real Time DI** page.

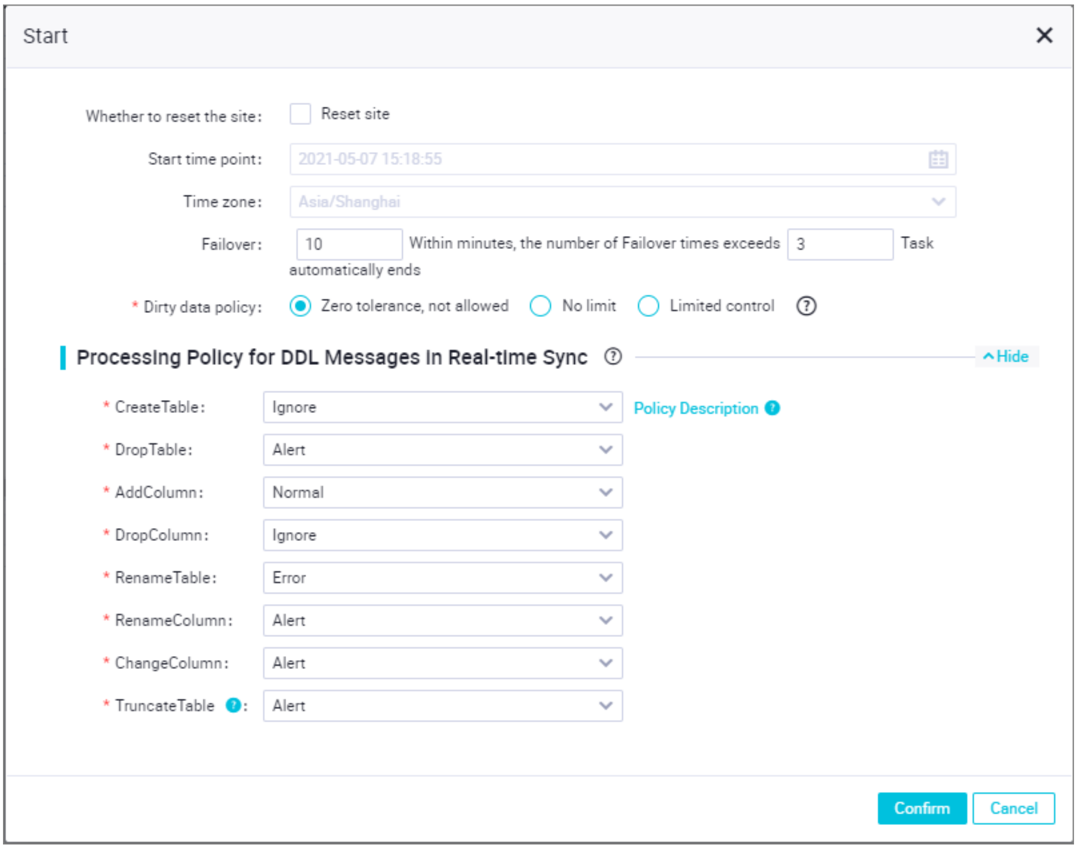
2. View the details of a real-time data sync node.

   On the **Real Time DI** page, find the real-time data sync node that you want to view and click the node name.

3. Start the real-time data sync node.

   i. Go back to the previous page, find the real-time data sync node that you want to start and click **Start** in the **Operation**.

   ii. In the **Start** dialog box, set the parameters that are described in the following table.



Start

Whether to reset the site: □ Reset site

Start time point: 2021-05-07 15:18:55

Time zone: Asia/Shanghai

Failover: 10 Within minutes, the number of Failover times exceeds 3 Task automatically ends

* Dirty data policy: ● Zero tolerance, not allowed ○ No limit ○ Limited control

Processing Policy for DDL Messages in Real-time Sync

* CreateTable: Ignore

* DropTable: Alert

* AddColumn: Normal

* DropColumn: Ignore

* RenameTable: Error

* RenameColumn: Alert

* ChangeColumn: Alert

* TruncateTable: Alert

Parameter	Description
Whether to reset the site	Specifies whether to set the point in time for the next startup. If you select the Reset site parameter, the **Start time point** and **Time zone** parameters are required.
Start time point	The date and time for starting the real-time data sync node.
Time zone	The time zone in which the real-time data sync node is run. You can select a time zone from the **Time zone** drop-down list.

Parameter	Description
Failover	The maximum number of failovers allowed within the specified time range. ⓘ Note If this parameter is not specified, the system automatically stops the node if the number of failovers exceeds 100 within 5 minutes. This prevents excessive resource consumption caused by the frequent starting of the node.
Dirty data policy	■ **Zero tolerance, not allowed**: The real-time sync node is automatically stopped if the node contains dirty data. ■ **No limit**: The real-time data sync node can normally run regardless of whether the node contains dirty data. ■ **Limited control**: The real-time data sync node is automatically stopped if the amount of dirty data contained in the node exceeds a specified value.
Processing Policy for DDL Messages in Real-time Sync	You can modify the configured rules that are used to process DDL messages based on your business requirements. For more information, see Step 10 of this topic.

Manage the real-time data sync node

- Stop a real-time data sync node that is running.

  Find the real-time data sync node that you want to stop and click **Stop** in the Operation column. In the message that appears, click **Stop**.

- Undeploy a real-time data sync node that is not running.

  Find the real-time data sync node that you want to undeploy and click **Undeploy** in the Operation column. In the message that appears, click **Undeploy**.

- View the alert information of a real-time data sync node.

  Find the real-time data sync node that you want to view and click **Alert settings** in the Actions column. In the **Alert settings** dialog box, view the alert events and alert rules.

- Configure alert rules for a real-time data sync node.

  i. Find the real-time data sync node for which you want to configure alert rules and click **Configure Alert Rule** in the lower part of the **Real Time DI** page.

  ii. In the **New rule** dialog box, set the parameters that are described in the following table.

Parameter	Description
Name	The name of the alert rule.
Description	The description of the alert rule.
Indicators	The metric for which an alert is reported. Valid values: ■ **Status** ■ **Business delay** ■ **Failover** ■ **Dirty Data** ■ **Not Supported by DDL Statement**
Threshold	The threshold for reporting an alert. Specify the **WARNING In** and **CRITICAL In** parameters. The default values of the parameters are 5 minutes.
Alarm interval	The interval at which an alert is reported. The default value is 5 minutes.

Parameter	Description
WARNING	The method that is used to send alert notifications. You can specify one or more methods. Valid values: **Mail**, **SMS**, and **DingTalk**. **Note** Only Singapore, Malaysia(Kuala Limpur), and Germany(Frankfurt) support the SMS reminding method. To use the SMS reminding method in other regions, submit a ticket to contact DataWorks technical support.
CRITICAL	
Receiver (Non-DingTalk)	The recipient of alert notifications.

iii. Click **Confirm**.

- Modifies alert rules for real-time data sync nodes at a time.

  i. Select one or more real-time data sync nodes for which you want to modify alert rules and click **Operation alarm** in the lower part of the **Real Time DI** page.

  ii. In the **Operation alarm** dialog box, modify the values of the **Type** and **Indicators** parameters.

  iii. Click **Confirm**.

3.6. Synchronize all data in a database to Hologres

3.6.1. Plan and configure resources

When you use DataWorks to synchronize data, you can use only exclusive resource groups for Data Integration to run real-time synchronization nodes. This topic describes the resources and configurations required to run real-time synchronization nodes.

Context

- Resource planning and preparation

  Before you use a data synchronization node to synchronize data, you must purchase an exclusive resource group for data integration and add the resource group to DataWorks for subsequent use.

  For more information about exclusive resource groups for Data Integration, see Exclusive resources for Data Integration.

- Network connections

  An exclusive resource group for Data Integration is essentially a group of Elastic Compute Service (ECS) instances. After you purchase and create such an exclusive resource group, it is isolated from other services. You must associate the resource group with a virtual private cloud (VPC) to ensure network connectivity between the resource group and data sources during subsequent data synchronization.

What's next

After you plan and configure resources, you can configure data sources. You must configure network connections for the data sources and permissions to access the data sources. This facilitates the creation of a real-time synchronization node. You can synchronize data only from PolarDB, Oracle, MySQL data sources to Hologres. You can select a data source based on your business requirements. For more information about how to configure a data source, see Configure a data source (PolarDB), Configure data sources for data synchronization from Oracle, Configure data sources for data synchronization from MySQL.

3.6.2. Configure a data source (PolarDB)

If you want to synchronize data from PolarDB to Hologres, the source is PolarDB, and the destination is Hologres. Before you run a data synchronization task, you must refer to the operations in this topic to prepare the configurations, such as network environments and whitelists, for both the source and destination.

Prerequisites

Before you configure a data source, make sure that the following operations are performed:

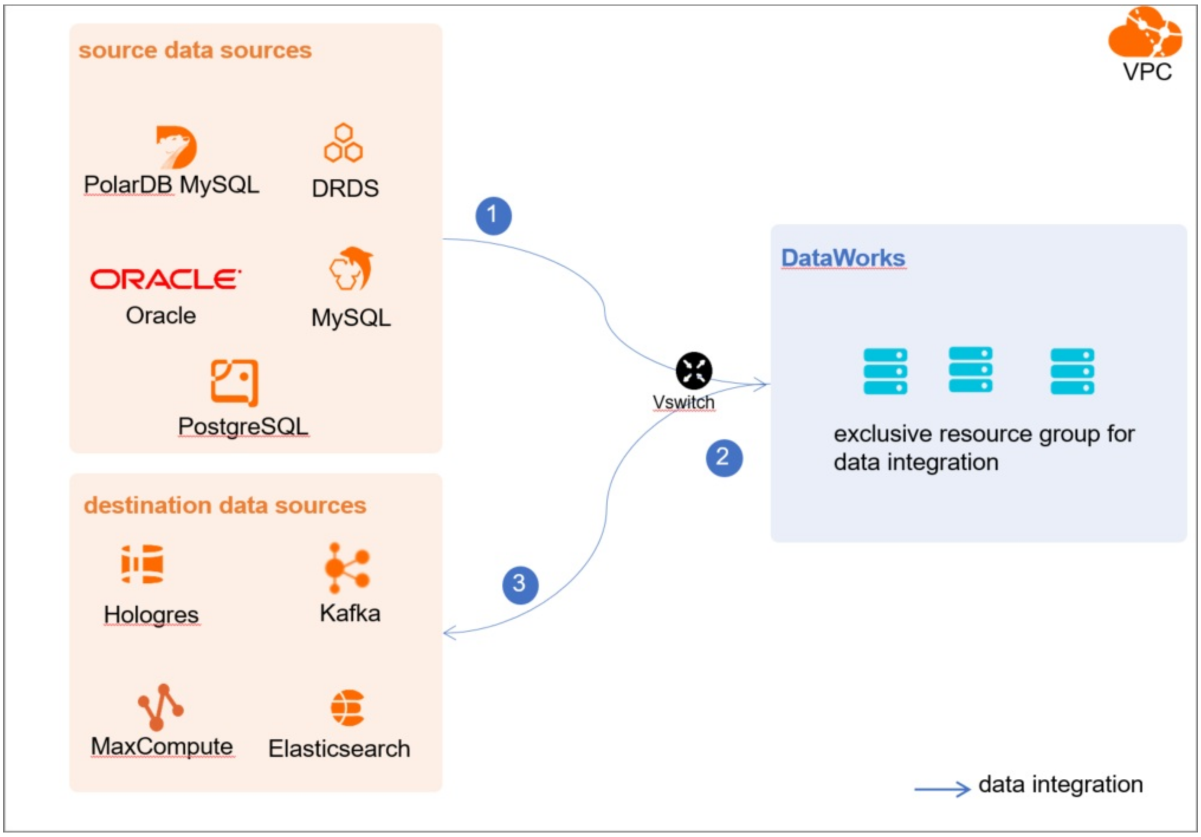
- Prepare data sources: A PolarDB for MySQL cluster and a Hologres instance are created. In this topic, a PolarDB for MySQL cluster is used as the source.
- Plan and prepare resources: An exclusive resource group for data integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
    - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
    - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

  If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source is a PolarDB for MySQL cluster, you must enable the binary logging feature for the cluster. PolarDB for MySQL is fully compatible with MySQL and uses high-level physical logs to replace binary logs. To facilitate the integration between PolarDB and the MySQL ecosystem, you can enable the binary logging feature for PolarDB clusters.

Limits

- Only PolarDB for MySQL clusters can be used as sources in data synchronization solutions. Other types of PolarDB data sources are not supported. In this topic, PolarDB indicates PolarDB for MySQL data sources.
- Only data stored on the primary node of a PolarDB for MySQL cluster can be synchronized.

Procedure

1. Configure a whitelist for the PolarDB for MySQL cluster.

   To add the CIDR block of the VPC where the exclusive resource group for Data Integration resides to a whitelist of the PolarDB for MySQL cluster, perform the following steps:

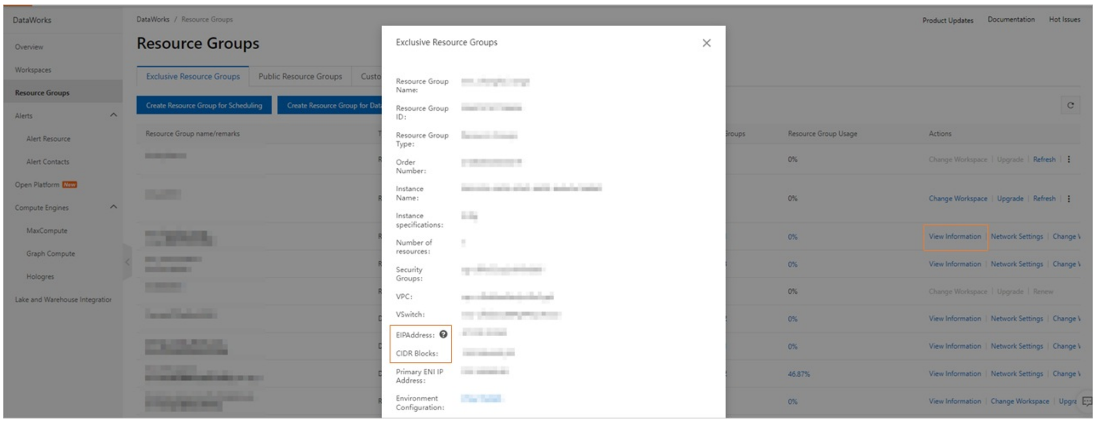
   i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

      a. Log on to the DataWorks console.

      b. In the left-side navigation pane, click **Resource Groups**.

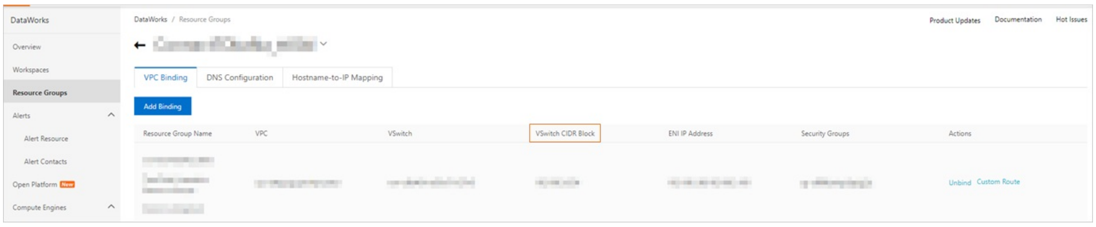
      c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

      d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.

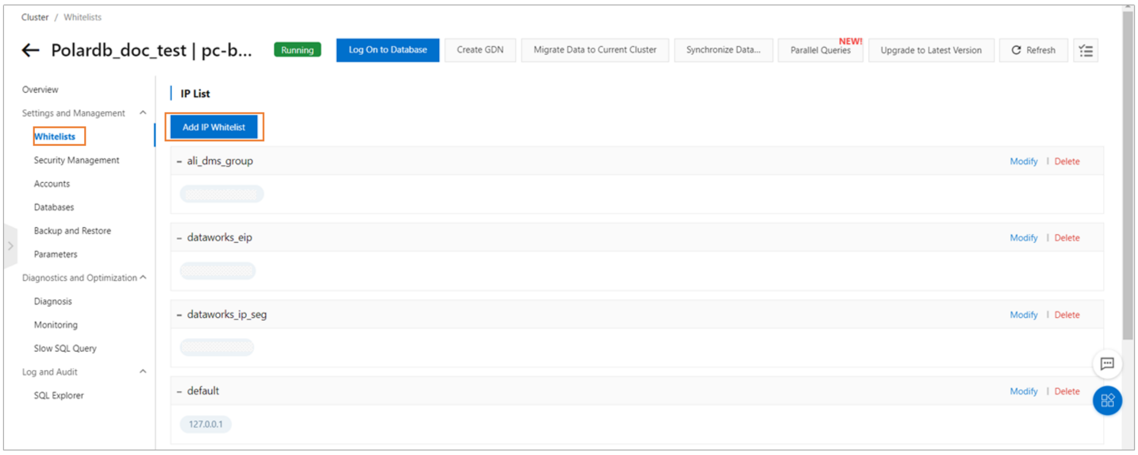


      e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

      f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR blocks recorded in the preceding steps to the whitelist of the PolarDB for MySQL cluster.



For more information, see Configure an IP whitelist.

2. Create an account and grant the required permissions to the account.

   You must create an account to log on to the database of the PolarDB for MySQL cluster. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

   i. Create an account.

   For more information, see Create a database account.

   ii. Grant the required permissions to the account.

   You can run the following command to grant the required permissions to the account, or you can directly assign the `SUPER` role to the account.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Account for data synchronization';
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%';
```

3. Enable the binary logging feature for the PolarDB for MySQL cluster.

   For more information, see Enable binary logging.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

3.6.3. Configure data sources for data synchronization from Oracle

Before you synchronize data from Oracle to Hologres in real time, you can refer to the operations described in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions. You must configure a source Oracle data source and a destination Hologres data source.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

-
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.

  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

3.6.4. Configure data sources for data synchronization from MySQL

Before you synchronize data from MySQL to Hologres in real time, you can refer to the operations described in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions. You must configure a source MySQL data source and a destination Hologres data source.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

- Prepare data sources: A source MySQL data source and a destination Hologres data source are created.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.
- Execute the following statement to check the version of your MySQL database. You can use a `MySQL 5.x` or `MySQL 8.x` database.

  ```
  select version();
  ```

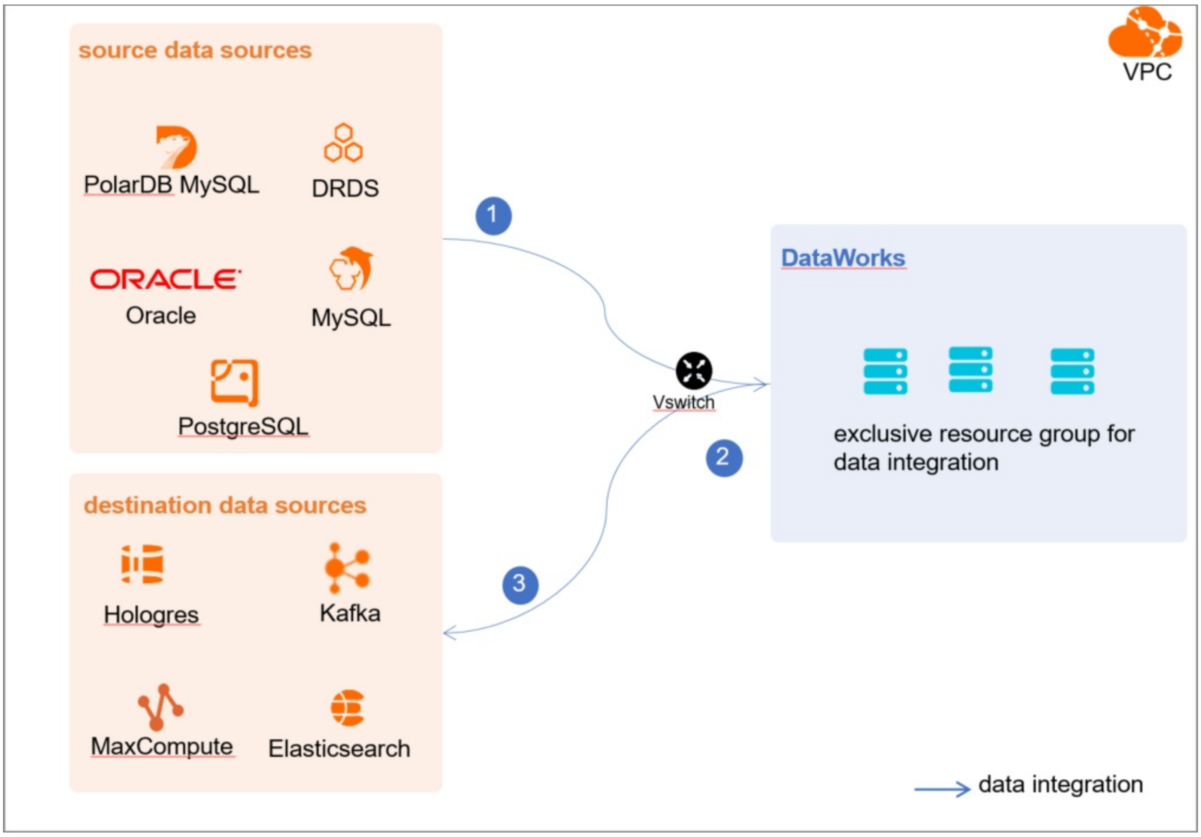
  > **Note** Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.x` or `V8.x`. PolarDB for MySQL is not supported. If the MySQL version of your ApsaraDB RDS for MySQL database is not `V5.x` or `V8.x`, use an ApsaraDB RDS for MySQL database whose MySQL version is `V5.x` or `V8.x`. Otherwise, the data synchronization node fails to run.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source data source is a MySQL database, you must enable the binary logging feature. Binary logs record changes to all table schemas and modifications on table data. You can execute statements such as CREATE and ALTER to perform operations on table schemas. You can execute statements such as INSERT, UPDATE, and DELETE to perform operations on table data. You can use binary logs to view the change history of the database, back up incremental data and restore data in the database, and replicate data from the primary database to secondary databases.

  Formats of binary logs:

  - Statement: SQL statement-based replication. Binary logs in this format record the SQL statements that are executed to modify data entries.
  - Row: row-based replication. Binary logs in this format record only modification details about data entries in rows.
  - Mixed: replication in mixed mode. This mode combines the statement and row formats. In most cases, binary logs in the statement format are used to record the SQL statements that are executed to modify data entries, such as functions. If data replication from the primary database to secondary databases cannot be implemented by using binary logs in this format, switch to the row format. MySQL determines which format to use based on each SQL statement that is executed.

Limits

Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.x` or `V8.x` . PolarDB for MySQL is not supported.

Procedure

1. Configure a whitelist for the MySQL database.

Add the CIDR block of the VPC in which the exclusive resource group resides to the whitelist of the MySQL database.

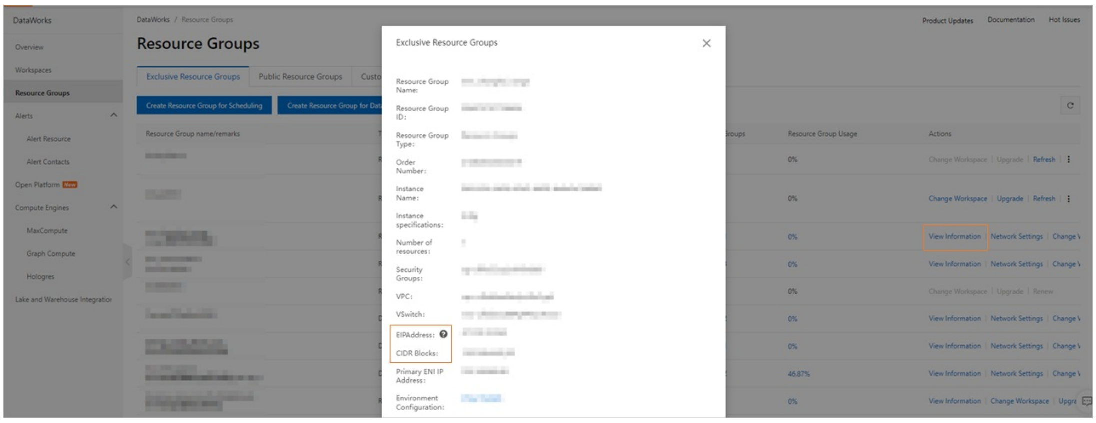
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

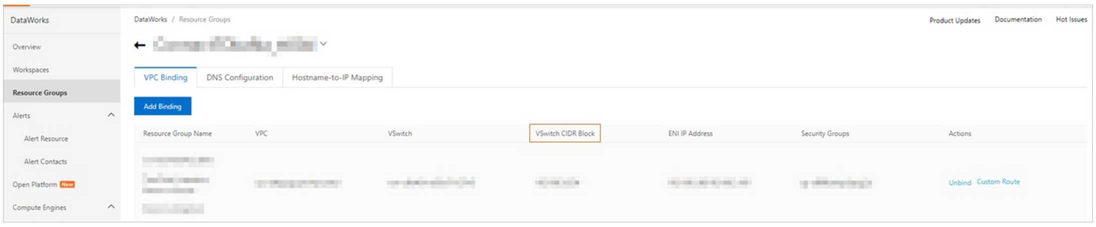
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR block of the exclusive resource group to the whitelist of the MySQL database.

2. Create an account and grant permissions to the account.

You must create an account to log on to the MySQL database. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

i. Create an account.

For more information, see Create an account to access a MySQL database.

ii. Grant permissions to the account.

You can run the following command to grant permissions to the account. Alternatively, you can grant the `SUPER` permission to the account. Replace `Account for data synchronization` with the created account when you execute the specific statement.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Password'; // Create an account that is used to synchronize data and set a password so that you can use the account and password to access the database from a host. % indicates a host.
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%'; // Grant the SELECT, REPLICATION SLAVE, and REPLICATION CLIENT permissions to the account.
```

`*.*` indicates that the synchronization account is granted the preceding permissions on all tables in all databases. You can also grant the preceding permissions on specified tables in the specified database to the synchronization account. For example, to grant the account the preceding permissions on the *user* table in the *test* database, execute the following statement: `GRANT SELECT, REPLICATION CLIENT ON test.user TO 'Account for data synchronization'@'%';` .

> **Note** The `REPLICATION SLAVE` permission is a global permission. You cannot grant this permission on specified tables in the specified database to the synchronization account.

3. Enable the binary logging feature for the MySQL database.

   Perform the following steps to check whether the binary logging feature is enabled and to query the format of binary logs:

   - Execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_bin";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled.

   - If you use a secondary database to synchronize data, execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_slave_updates";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled for the secondary database.

   If the returned result is different from the preceding result, follow the instructions described in the *MySQL documentation* to enable the binary logging feature.

   Execute the following statement to view the format of binary logs:

   ```
   show variables like "binlog_format";
   ```

   Returned result:

   - *ROW*: The format of binary logs is row.
   - *STATEMENT*: The format of binary logs is statement.
   - *MIXED*: The format of binary logs is mixed.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

3.6.5. Add a data source

Before you configure a data synchronization node to synchronize data from a data source to the Hologres data source, you must add both data sources to DataWorks for subsequent source and destination configurations.

Prerequisites

Before you add data sources, make sure that the following operations are performed:

- Prepare data sources: A source data source and a destination data source are created.
- Create and grant permission to an account: An account that is used to access data sources is created.

Precautions

DataWorks provides workspaces in basic mode and standard mode. A workspace in basic mode does not isolate the development environment from the production environment. A workspace in standard mode isolates the development environment from the production environment.

If you use a workspace in standard mode, you must separately add data sources to the development environment and production environment.

Add a source PolarDB for MySQL data source

To add a source PolarDB for MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source PolarDB for MySQL data source, see Add a PolarDB data source.

If the source PolarDB for MySQL data source that you want to add fails the connectivity test, follow the instructions described in What do I do if the PolarDB data source cannot be connected? to handle the exception.

Add a source Oracle data source

To add a source Oracle data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source Oracle data source, see Add an Oracle data source.

Add a source MySQL data source

To add a source MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source MySQL data source, see Add a MySQL data source.

Add a destination Hologres data source

For more information about how to add a destination Hologres data source, see Add a Hologres data source.

What's next

After you add data sources, you can create and run a data sync node to synchronize data from the source data source to the destination data source.

For more information, see Configure and manage a real-time sync node.

3.6.6. Configure and manage a real-time sync node

After you prepare data sources, network environments, and resources, you can create a real-time sync node to synchronize data to Hologres. This topic describes how to create a real-time sync node and view the status of the node.

Prerequisites

Before you create a real-time sync node, read the following topics to make sure that the required operations are performed:

- Plan and configure resources
- Configure a data source (PolarDB)
- Configure data sources for data synchronization from Oracle
- Add a data source

Limits

- DataWorks allows you to use only exclusive resource groups for Data Integration to run real-time data sync nodes.
- You can run real-time sync nodes to synchronize data only from PolarDB, Oracle, MySQL, or SQL Server data sources to Hologres.
- A real-time data sync node cannot be used to synchronize data in a table that has no primary key.

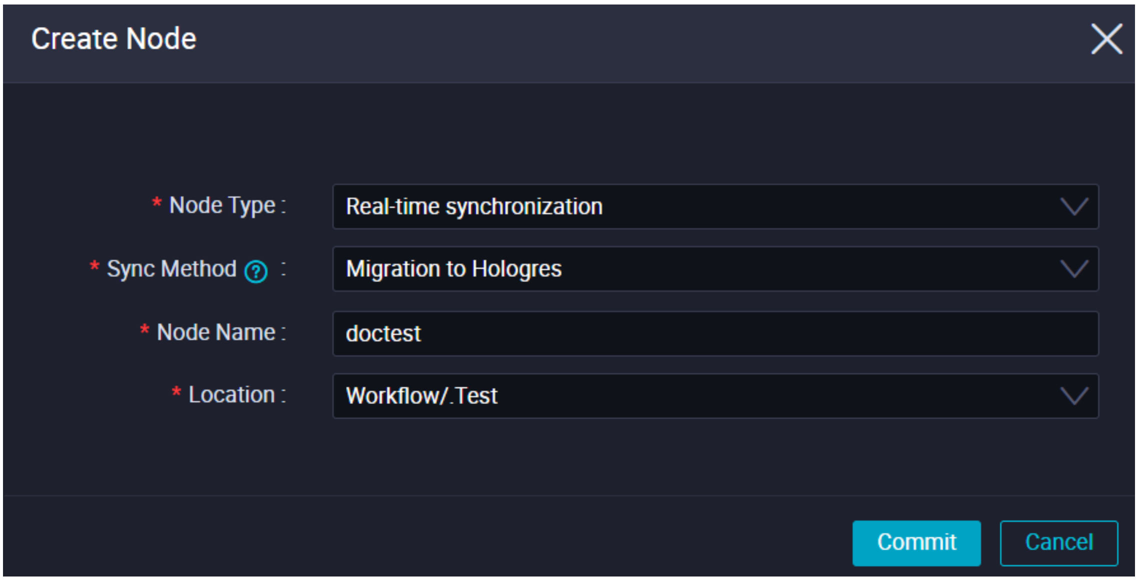
Create a real-time sync node

1. Log on to the DataWorks console.
2. In the left-side navigation pane, click **Workspaces**.
3. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
4.
5. Create a real-time sync node.

   i. On the DataStudio page, move the pointer over the **+Create** icon and choose **Data Integration > Real-time synchronization**.

   Alternatively, find the workflow in which you want to create a real-time data sync node and right-click the **Data Integration**. From the shortcut menu, choose **Create > Real-time synchronization**.

   ii. In the **Create Node** dialog box, set the parameters that are described in the following table.



Create Node

* Node Type : Real-time synchronization

* Sync Method : Migration to Hologres

* Node Name : doctest

* Location : Workflow/.Test

Parameter	Description
Node Type	The type of the node. Default value: **Real-time synchronization**.
Sync Method	Set the parameter to **Migration to Hologres**. In this case, partial or all tables in your desired database are migrated to Hologres.
Node Name	The name of the node. The name must be 1 to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).
Location	The directory in which the real-time sync node is stored.

   iii. Click **Commit**. You are navigated to the configuration tab of the real-time data sync node.

6. Select a resource group.

   i. On the right side of the configuration tab, click the **Basic Configuration** tab.

   ii. In the panel that appears, select the resource group that you want to use from the **Resource Group** drop-down list.

   > Note
   >
   > DataWorks allows you to use only exclusive resource groups for Data Integration to run real-time data sync nodes.
   >
   > If no exclusive resource group for Data Integration exists, click **Create Exclusive Resource Group for Data Integration** to create a resource group. For more information, see Overview.

7. Select a source and configure synchronization rules.

i. In the **Data Source** section, set the **Type** and **Data source** parameters.

> **Note** You can set the Type parameter only to MySQL, SQL Server, Oracle, or PolarDB.

ii.

iii.

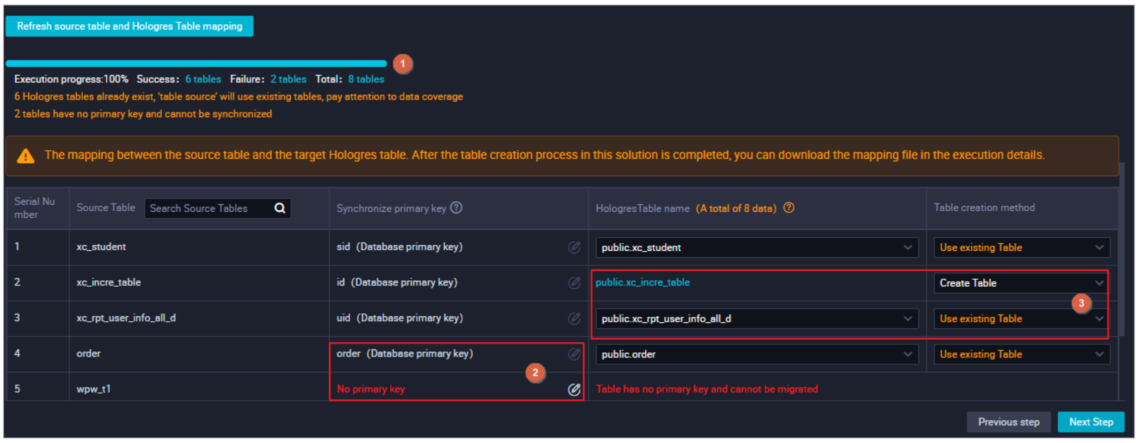
iv.

8. Select a data source as the destination and configure the formats for the destination tables.

i.

ii.

iii. View the mapping progress, source tables, and mapped destination tables.



Execution progress:100% Success: 6 tables Failure: 2 tables Total: 8 tables

6 Hologres tables already exist, 'table source' will use existing tables, pay attention to data coverage

2 tables have no primary key and cannot be synchronized

The mapping between the source table and the target Hologres table. After the table creation process in this solution is completed, you can download the mapping file in the execution details.

Serial Number	Source Table (Search Source Tables)	Synchronize primary key	HologresTable name (A total of 8 data)	Table creation method
1	xc_student	sid (Database primary key)	public.xc_student	Use existing Table
2	xc_incre_table	id (Database primary key)	public.xc_incre_table	Create Table
3	xc_rpt_user_info_all_d	uid (Database primary key)	public.xc_rpt_user_info_all_d	Use existing Table
4	order	order (Database primary key)	public.order	Use existing Table
5	wpw_t1	No primary key	Table has no primary key and cannot be migrated	

No.	Description
1	The progress of mapping the source tables to the destination tables. **Note** The mapping may require a long period of time if data is synchronized from a large number of tables.
2	■ If the tables in the source database contain primary keys, the system removes duplicate data based on the primary keys during the synchronization. ■ If the tables in the source database do not contain primary keys, you can click the ✎ icon to customize primary keys. You can use one field or a combination of several fields as the primary keys of the tables. This way, the system removes duplicate data based on the primary keys during the synchronization.
3	The method that is used to create a table. Valid values: **Create Table** and **Use existing Table**.

iv. Click **Next Step**.

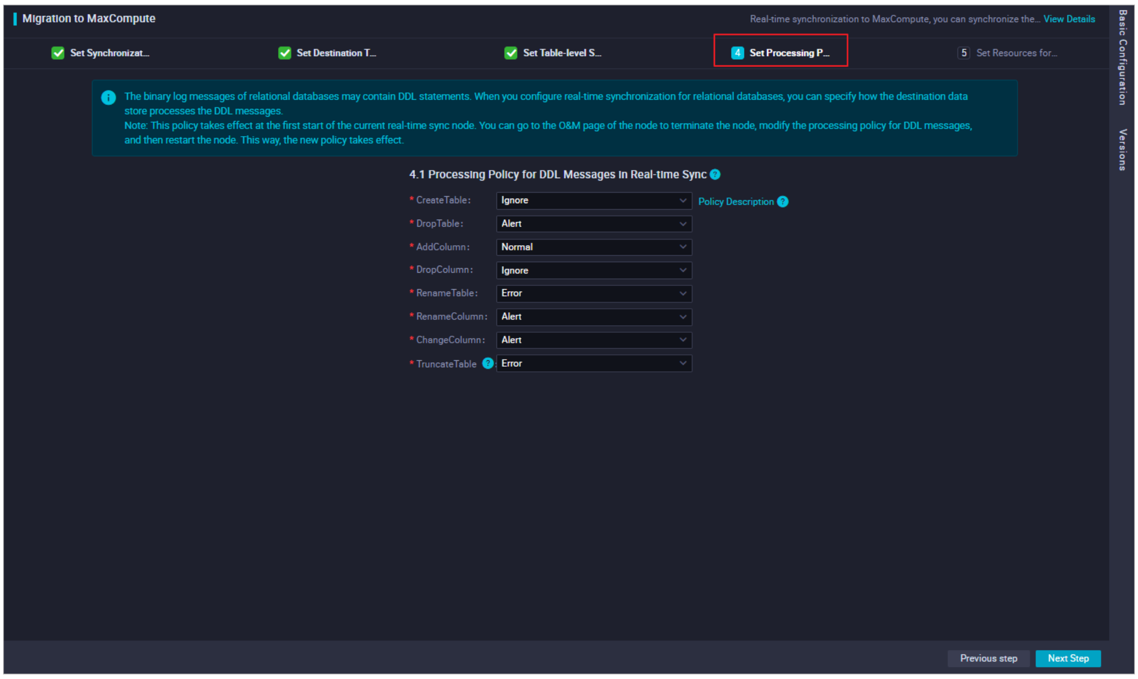
If you set the **Table creation method** to **Create Table**, you must click **Start table building** in the **Create tables automatically** dialog box to create destination Hologres tables.

9. Configure rules for processing DDL messages.

DDL statements exist in the source. Before you synchronize data, you can configure synchronization rules for different DDL statements based on your business requirements.

> Note The rules apply when a real-time sync node is run for the first time. If you want to modify the rules in subsequent operations, go to the Real Time DI page. For more information, see the "*Start the real-time sync node*" section of this topic.

i. In the **Set Processing Policy for DDL Messages** step, configure rules to process DDL messages during data synchronization.



Migration to MaxCompute

Real-time synchronization to MaxCompute, you can synchronize the... View Details

The binary log messages of relational databases may contain DDL statements. When you configure real-time synchronization for relational databases, you can specify how the destination data store processes the DDL messages.
Note: This policy takes effect at the first start of the current real-time sync node. You can go to the O&M page of the node to terminate the node, modify the processing policy for DDL messages, and then restart the node. This way, the new policy takes effect.

4.1 Processing Policy for DDL Messages in Real-time Sync

- * CreateTable: Ignore
- * DropTable: Alert
- * AddColumn: Normal
- * DropColumn: Ignore
- * RenameTable: Error
- * RenameColumn: Alert
- * ChangeColumn: Alert
- * TruncateTable: Error

The following table describes the processing rules for different DDL messages.

DDL message	Rule
CreateTable	DataWorks processes a DDL message of the related type based on the following rules after it receives the message: ■ **Normal**: sends the message to the destination. Then, the destination processes the message. Each destination may process DDL messages based on its own business logic. If you select Normal for CreateTable, DataWorks only forwards the messages. ■ **Ignore**: ignores the message and does not send it to the destination. ■ **Alert**: ignores the message and records the alert in real-time synchronization logs. In addition, the alert contains information about the reason indicating that a message is ignored because of a running error. ■ **Error**: returns an error when the real-time sync solution is running and terminates the real-time sync solution.
DropTable	
AddColumn	
DropColumn	
RenameTable	
RenameColumn	
ChangeColumn	
TruncateTable	

ii. Click **Next**.

10. Configure the resources required by the data sync node.

i. In the **Set Resources for Solution Running** step, set the parameters that are described in the following table.

Parameter	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: **15**.
Maximum number of parallel threads allowed to read by destination	The maximum number of parallel threads that the sync node uses to read data from the source table or write data to the destination. Maximum value: 32. Specify an appropriate number based on the resources of the source and the destination.

ii. Click **Complete Configuration**.

Commit and deploy the real-time data sync node

1.

Start the real-time sync node

1. Go to the Operation Center page.

   After you commit and deploy the real-time data sync node, click **Operation Center** in the upper-right corner of the DataStudio page to manage the node on the **Real Time DI** page.

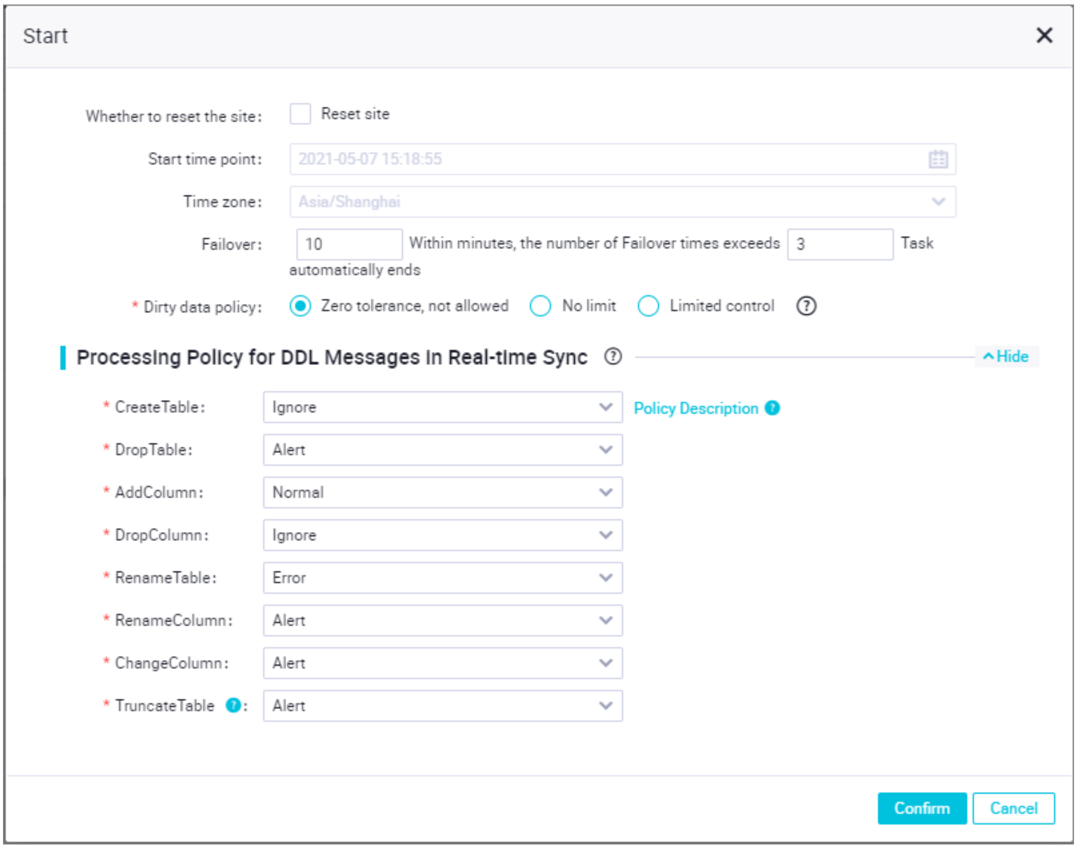
2. View the details of a real-time data sync node.

   On the **Real Time DI** page, find the real-time data sync node that you want to view and click the node name.

3. Start the real-time sync node.

   i. Go back to the previous page and click **Start** in the **Operation** column that corresponds to your desired node.

ii. In the **Start** dialog box, set the parameters as required.



Start

Whether to reset the site: □ Reset site

Start time point: 2021-05-07 15:18:55

Time zone: Asia/Shanghai

Failover: 10 Within minutes, the number of Failover times exceeds 3 Task automatically ends

* Dirty data policy: ● Zero tolerance, not allowed ○ No limit ○ Limited control

Processing Policy for DDL Messages in Real-time Sync

* CreateTable: Ignore

* DropTable: Alert

* AddColumn: Normal

* DropColumn: Ignore

* RenameTable: Error

* RenameColumn: Alert

* ChangeColumn: Alert

* TruncateTable: Alert

Parameter	Description
Whether to reset the site	Specifies whether to set the point in time for the next startup. If you select Reset site, the **Start time point** and **Time zone** parameters are required.
Start time point	The date and time for starting the real-time sync node.
Time zone	The time zone in which the real-time sync node is run. You can select a time zone from the **Time zone** drop-down list.
Failover	The maximum number of failovers allowed within the specified time range. **Note** If this parameter is not specified, the system automatically stops the node if the number of failovers exceeds 100 within 5 minutes. This avoids excessive resource consumption caused by the frequent starting of the node.
Dirty data policy	■ **Zero tolerance, not allowed**: The real-time sync node is automatically stopped if the node contains dirty data. ■ **No limit**: The real-time sync node can normally run regardless of whether the node contains dirty data. ■ **Limited control**: The real-time sync node is automatically stopped if the amount of dirty data contained in the node exceeds a specified value.
Processing Policy for DDL Messages in Real-time Sync	You can modify the configured rules that are used to process DDL messages based on your business requirements. For more information, see Step 10 in this topic.

Manage the real-time data sync node

- Stop a real-time data sync node that is running.

  Find the real-time data sync node that you want to stop and click **Stop** in the Operation column. In the message that appears, click **Stop**.

- Undeploy a real-time data sync node that is not running.

  Find the real-time data sync node that you want to undeploy and click **Undeploy** in the Operation column. In the message that appears, click **Undeploy**.

- View the alert information of a real-time data sync node.

  Find the real-time data sync node that you want to view and click **Alert settings** in the Actions column. In the **Alert settings** dialog box, view the alert events and alert rules.

- Configure alert rules for a real-time data sync node.

  i. Find the real-time data sync node for which you want to configure alert rules and click **Configure Alert Rule** in the lower part of the **Real Time DI** page.

  ii. In the **New rule** dialog box, set the parameters that are described in the following table.

Parameter	Description
Name	The name of the alert rule.
Description	The description of the alert rule.
Indicators	The metric for which an alert is reported. Valid values: ▪ **Status** ▪ **Business delay** ▪ **Failover** ▪ **Dirty Data** ▪ **Not Supported by DDL Statement**
Threshold	The threshold for reporting an alert. Specify the **WARNING In** and **CRITICAL In** parameters. The default values of the parameters are 5 minutes.
Alarm interval	The interval at which an alert is reported. The default value is 5 minutes.
WARNING	The method that is used to send alert notifications. You can specify one or more methods. Valid values: **Mail**, **SMS**, and **DingTalk**. **Note** Only Singapore, Malaysia(Kuala Limpur), and Germany(Frankfurt) support the SMS reminding method. To use the SMS reminding method in other regions, submit a ticket to contact DataWorks technical support.
CRITICAL	
Receiver (Non-DingTalk)	The recipient of alert notifications.

  iii. Click **Confirm**.

- Modifies alert rules for real-time data sync nodes at a time.

  i. Select one or more real-time data sync nodes for which you want to modify alert rules and click **Operation alarm** in the lower part of the **Real Time DI** page.

  ii. In the **Operation alarm** dialog box, modify the values of the **Type** and **Indicators** parameters.

  iii. Click **Confirm**.

3.6.7. FAQ

This topic provides answers to some commonly asked questions about data synchronization to Hologres.

- What do I do if the connectivity test fails for the PolarDB data source?
- What do I do if the connectivity test fails for the Oracle data source?
- What do I do if the connectivity test fails for the MySQL data source?

- The system displays the following error message for a real-time data synchronization node: com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX. What do I do?
- The system displays the following error message for a real-time data synchronization node: com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation. What do I do?
- The system displays the following error message for a real-time data synchronization node: com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first. What do I do?

What do I do if the connectivity test fails for the PolarDB data source?

- Problem description: The connectivity test fails when I add a PolarDB data source.
- Solution: Set Data source type to Connection string mode and check the whitelist configuration of the PolarDB cluster and the virtual private cloud (VPC) configuration of your exclusive resource group.

What do I do if the connectivity test fails for the Oracle data source?

- Problem description: The connectivity test fails when I add an Oracle data source.
- Solution: Set Data source type to Connection string mode and check the whitelist configuration of the PolarDB cluster and the virtual private cloud (VPC) configuration of your exclusive resource group.

What do I do if the connectivity test fails for the MySQL data source?

- Problem description: The connectivity test fails when I add a MySQL data source.
- Solution: Set Data source type to Connection string mode and check the whitelist configuration of the PolarDB cluster and the virtual private cloud (VPC) configuration of your exclusive resource group.

The system displays the following error message for a real-time data synchronization node: com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX. What do I do?

- Problem description: The real-time synchronization node fails to run, and the system displays the error message " `com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX` ."
- Cause: The binary logging feature is disabled for the PolarDB data source.
- Solution: Enable the binary logging feature for the PolarDB data source. For more information, see Configure a data source (PolarDB). Change one or more data records and change the start time to run the real-time data synchronization node to the current time.

The system displays the following error message for a real-time data synchronization node: com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation. What do I do?

- Problem description: The real-time synchronization node fails to run, and the system displays the error message " `com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation` ."
- Cause: The account used to synchronize data is not authorized to access the PolarDB data source, or the PolarDB database connected is not deployed on the PolarDB Writer node.
- Solution: Refer to the operations in Configure a data source (PolarDB) to authorize the account to access the PolarDB data source. Alternatively, check whether the PolarDB database connected is deployed on the primary node. During the running of a real-time data synchronization node, the system cannot capture data from the read-only nodes of the PolarDB cluster.

The system displays the following error message for a real-time data synchronization node: com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first. What do I do?

- Problem description: The real-time synchronization node fails to run, and the system displays the error message " `com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first` ."
- Cause: The loose_polar_log_bin parameter is not specified for the PolarDB data source.
- Solution: Specify the loose_polar_log_bin parameter. For more information, see Configure a data source (PolarDB).

3.7. Synchronize database data to AnalyticDB for MySQL

3.7.1. Plan and configure resources

When you use DataWorks to synchronize data, you can use only exclusive resource groups for data integration to run real-time sync nodes. This topic describes the resources and configurations required to run real-time sync nodes.

Context

- Resource planning and preparation

  Before you use a data synchronization node to synchronize data, you must purchase an exclusive resource group for data integration and add the resource group to DataWorks for subsequent use.

  For more information about exclusive resource groups for Data Integration, see Exclusive resources for Data Integration.

- Network connections

  An exclusive resource group for Data Integration is essentially a group of Elastic Compute Service (ECS) instances. After you purchase and create such an exclusive resource group, it is isolated from other services. You must associate the resource group with a virtual private cloud (VPC) to ensure network connectivity between the resource group and data sources during subsequent data synchronization.

What's next

After you plan and configure resources, you can configure data sources. You must configure network connectivity for the data sources and permissions to access the data sources. This facilitates the creation of a real-time sync node. You can synchronize data to an AnalyticDB for MySQL data source only from a PolarDB or MySQL data source. You can select a PolarDB or MySQL data source based on your business requirements. For more information about how to configure a PolarDB or MySQL data source, see Configure a source PolarDB data source or Configure data sources for data synchronization from MySQL.

3.7.2. Configure a source PolarDB data source

Before you use DataWorks to synchronize data from a PolarDB data source to an AnalyticDB for MySQL data source, you must refer to the operations in this topic to prepare the configurations such as the network environments, whitelist, and account permissions for data synchronization.

Prerequisites

Before you configure a data source, make sure that the following operations are performed:

- Prepare data sources: A source PolarDB for MySQL cluster and a destination AnalyticDB for MySQL cluster are prepared. In this topic, a PolarDB for MySQL cluster is used as the source.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.

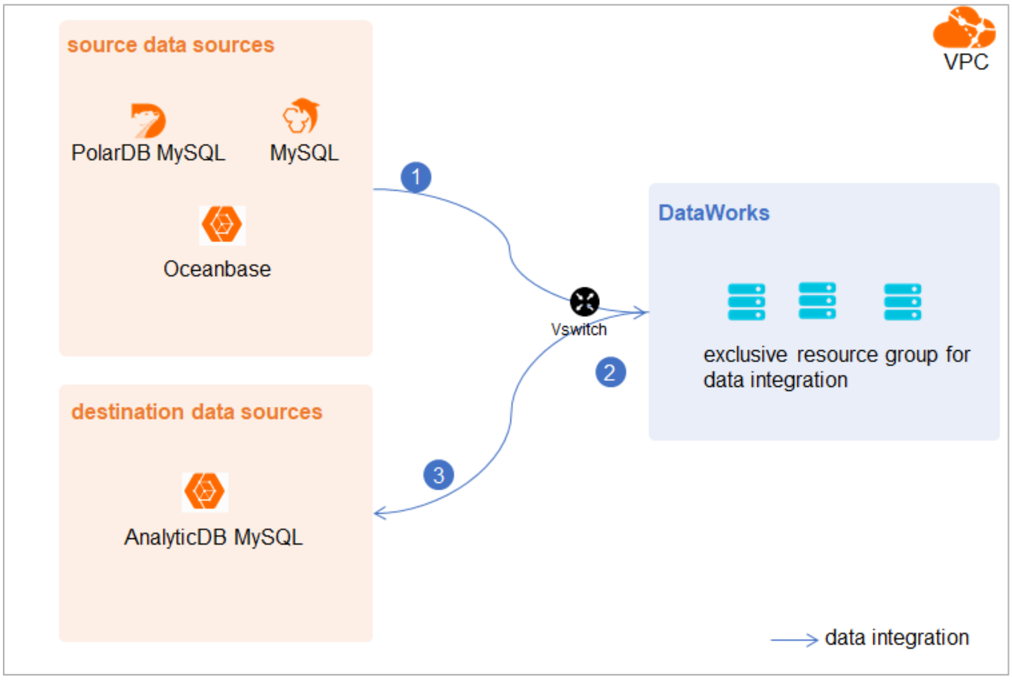
- If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
- If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and exclusive resource groups for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

  If the data sources and exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source is a PolarDB for MySQL cluster, you must enable the binary logging feature for the cluster. PolarDB for MySQL is fully compatible with MySQL and uses high-level physical logs to replace binary logs. To facilitate the integration between PolarDB and the MySQL ecosystem, you can enable the binary logging feature for PolarDB clusters.

Limits

- Only PolarDB for MySQL clusters can be used as sources in data synchronization solutions. Other types of PolarDB data sources are not supported. In this topic, PolarDB indicates PolarDB for MySQL data sources.
- Only data stored on the primary node of a PolarDB for MySQL cluster can be synchronized.

Procedure

1. Configure a whitelist for the PolarDB for MySQL cluster.

   To add the CIDR block of the VPC where the exclusive resource group for Data Integration resides to a whitelist of the PolarDB for MySQL cluster, perform the following steps:

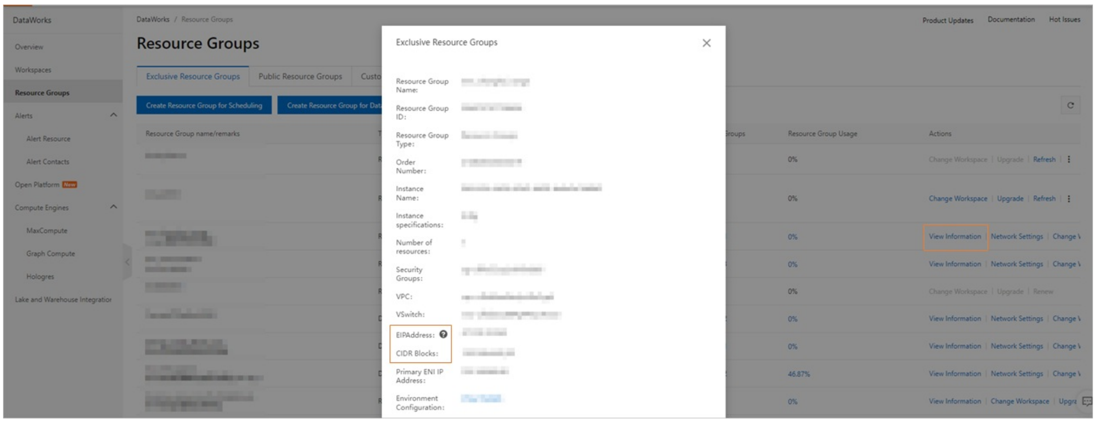
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

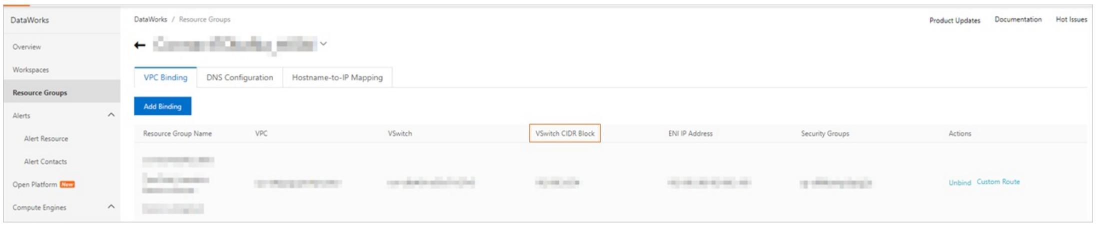
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.

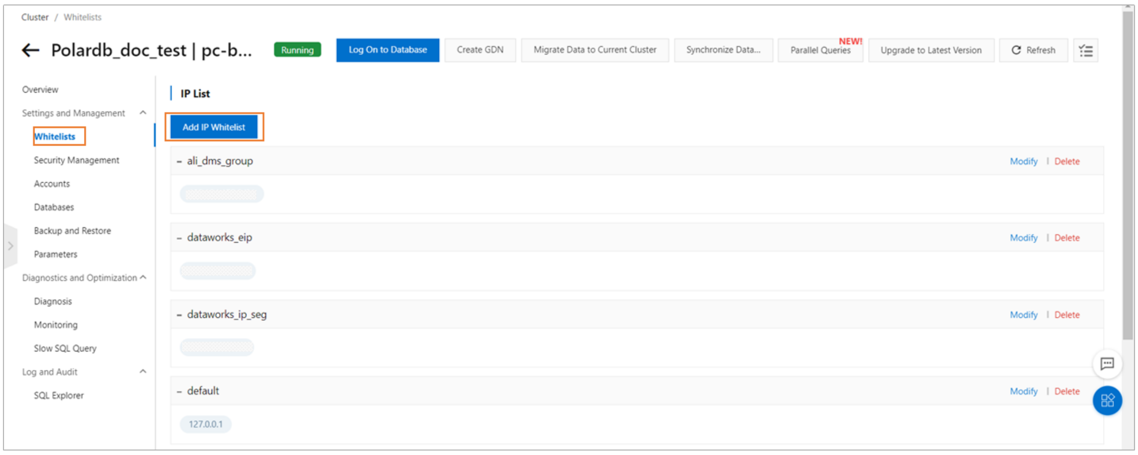


e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR blocks recorded in the preceding steps to the whitelist of the PolarDB for MySQL cluster.



For more information, see Configure an IP whitelist.

2. Create an account and grant the required permissions to the account.

You must create an account to log on to the database of the PolarDB for MySQL cluster. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

i. Create an account.

For more information, see Create a database account.

ii. Grant the required permissions to the account.

You can run the following command to grant the required permissions to the account, or you can directly assign the `SUPER` role to the account.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Account for data synchronization';
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%';
```

3. Enable the binary logging feature for the PolarDB for MySQL cluster.

   For more information, see Enable binary logging.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add data sources.

3.7.3. Configure data sources for data synchronization from MySQL

Before you synchronize data from MySQL to AnalyticDB for MySQL in real time, you can refer to the operations described in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions. You must configure a source MySQL data source and a destination AnalyticDB for MySQL data source.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

- Prepare data sources: A MySQL database and an AnalyticDB for MySQL instance are prepared.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.
- Execute the following statement to check the version of your MySQL database. You can use a `MySQL 5.X` or `MySQL 8.X` database.

  ```
  select version();
  ```

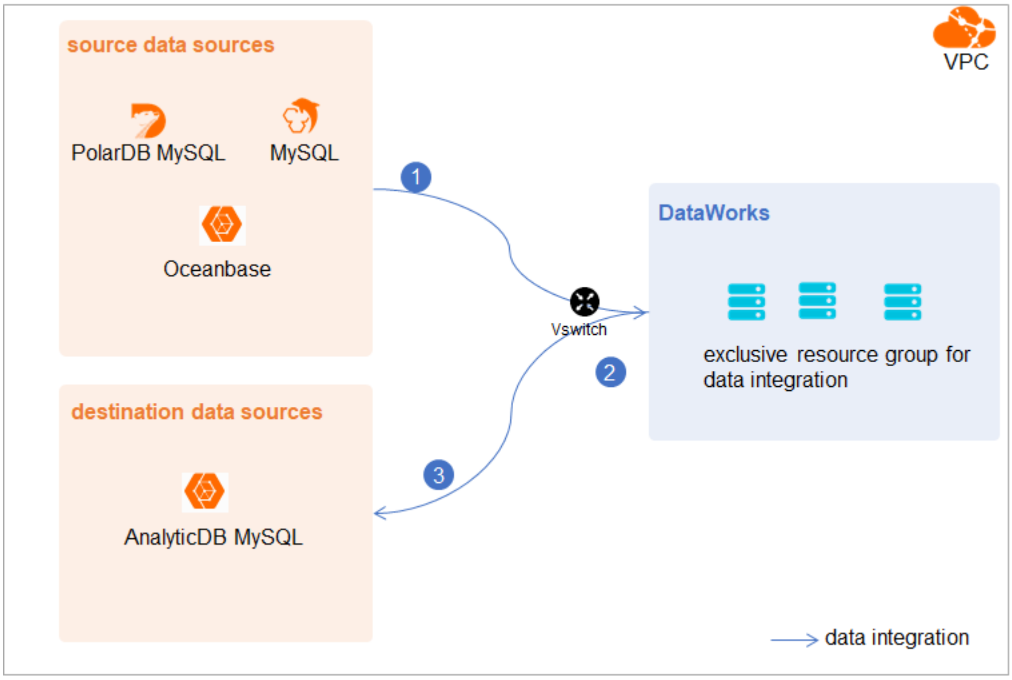
  > Note Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.X` or `V8.X`. PolarDB for MySQL is not supported. If the MySQL version of your ApsaraDB RDS for MySQL database is not `V5.X` or `V8.X`, use an ApsaraDB RDS for MySQL database whose MySQL version is `V5.X` or `V8.X`. Otherwise, the data synchronization node fails to run.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for data sources

If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source data source is a MySQL database, you must enable the binary logging feature. Binary logs record changes to all table schemas and modifications on table data. You can execute statements such as CREATE and ALTER to perform operations on table schemas. You can execute statements such as INSERT, UPDATE, and DELETE to perform operations on table data. You can use binary logs to view the change history of the database, back up incremental data and restore data in the database, and replicate data from the primary database to secondary databases.

  Formats of binary logs:

  - Statement: SQL statement-based replication. Binary logs in this format record the SQL statements that are executed to modify data entries.
  - Row: row-based replication. Binary logs in this format record only modification details about data entries in rows.
  - Mixed: replication in mixed mode. This mode combines the statement and row formats. In most cases, binary logs in the statement format are used to record the SQL statements that are executed to modify data entries, such as functions. If data replication from the primary database to secondary databases cannot be implemented by using binary logs in this format, switch to the row format. MySQL determines which format to use based on each SQL statement that is executed.

Limits

Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.x` or `V8.x`. PolarDB for MySQL is not supported.

Procedure

1. Configure a whitelist for the MySQL database.

   Add the CIDR block of the VPC in which the exclusive resource group resides to the whitelist of the MySQL database.

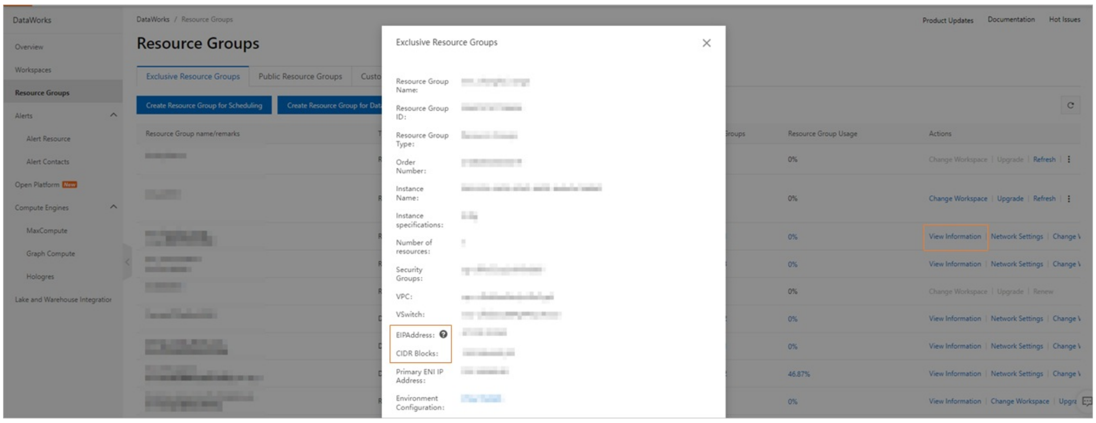
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

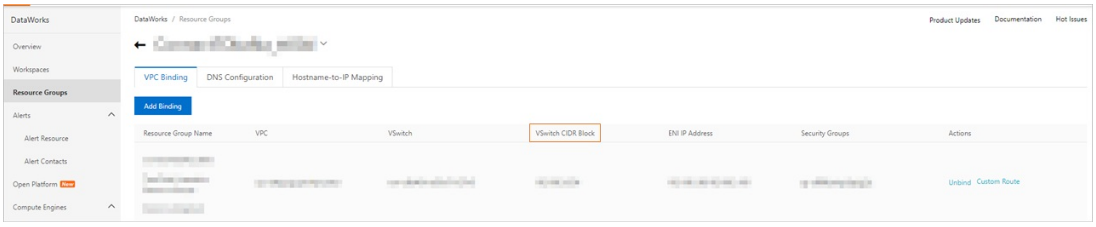
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR block of the exclusive resource group to the whitelist of the MySQL database.

2. Create an account and grant permissions to the account.

You must create an account to log on to the MySQL database. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

i. Create an account.

For more information, see Create an account to access a MySQL database.

ii. Grant permissions to the account.

You can run the following command to grant permissions to the account. Alternatively, you can grant the `SUPER` permission to the account. Replace `Account for data synchronization` with the created account when you execute the specific statement.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Password'; // Create an account that is used to synchronize data and set a password so that you can use the account and password to access the database from a host. % indicates a host.
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%'; // Grant the SELECT, REPLICATION SLAVE, and REPLICATION CLIENT permissions to the account.
```

`*.*` indicates that the synchronization account is granted the preceding permissions on all tables in all databases. You can also grant the preceding permissions on specified tables in the specified database to the synchronization account. For example, to grant the account the preceding permissions on the *user* table in the *test* database, execute the following statement: `GRANT SELECT, REPLICATION CLIENT ON test.user TO 'Account for data synchronization'@'%';` .

> **Note** The `REPLICATION SLAVE` permission is a global permission. You cannot grant this permission on specified tables in the specified database to the synchronization account.

3. Enable the binary logging feature for the MySQL database.

   Perform the following steps to check whether the binary logging feature is enabled and to query the format of binary logs:

   - Execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_bin";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled.

   - If you use a secondary database to synchronize data, execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_slave_updates";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled for the secondary database.

   If the returned result is different from the preceding result, follow the instructions described in the *MySQL documentation* to enable the binary logging feature.

   Execute the following statement to view the format of binary logs:

   ```
   show variables like "binlog_format";
   ```

   Returned result:

   - *ROW*: The format of binary logs is row.
   - *STATEMENT*: The format of binary logs is statement.
   - *MIXED*: The format of binary logs is mixed.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add data sources.

3.7.4. Configure a source ApsaraDB for OceanBase data source

Before you use DataWorks to synchronize data from an ApsaraDB for OceanBase cluster to an AnalyticDB for MySQL V3.0 cluster, you must refer to the operations in this topic to prepare the configurations such as the network environments, whitelist, and account permissions for data synchronization.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

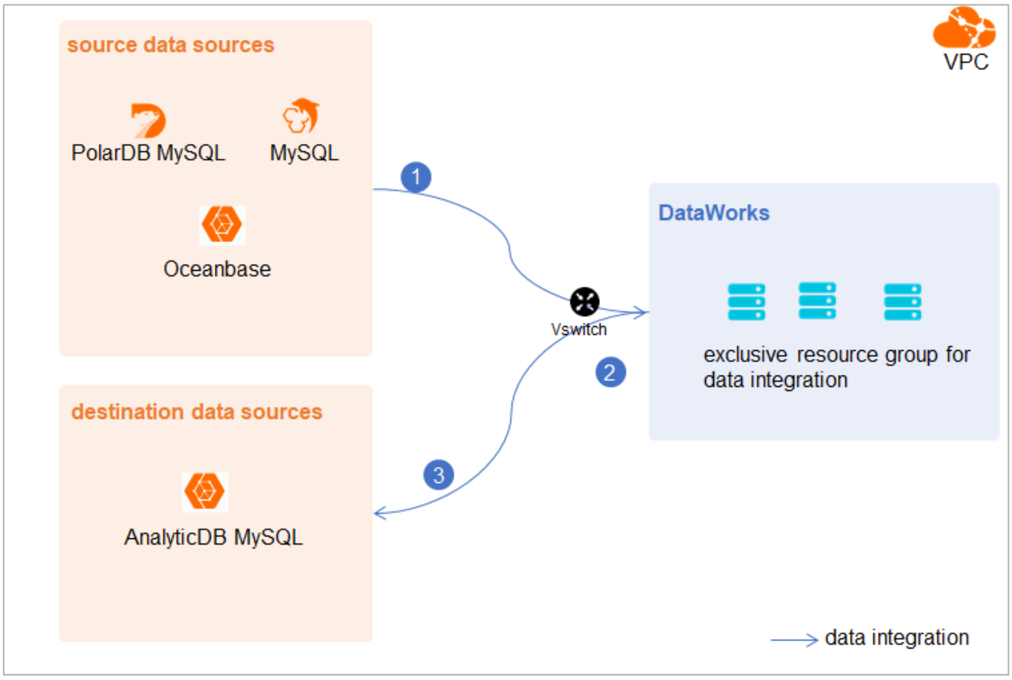
- Prepare data sources: An ApsaraDB for OceanBase cluster and a destination AnalyticDB for MySQL 3.0 cluster are prepared.
- Plan and prepare resources: An exclusive resource group for data integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
    - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
    - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources are connected to an exclusive resource group for Data Integration. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

  If the data sources and exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

Limits

ApsaraDB for OceanBase is a distributed relational database that can integrate data distributed in multiple physical databases into a unified logical database. However, you can synchronize data of only one physical ApsaraDB for OceanBase database to an AnalyticDB for MySQL cluster in real time.

Procedure

1. Configure a whitelist for the ApsaraDB for OceanBase cluster.

   Add the CIDR block of the VPC where the exclusive resource group resides to the whitelist of the ApsaraDB for OceanBase cluster.

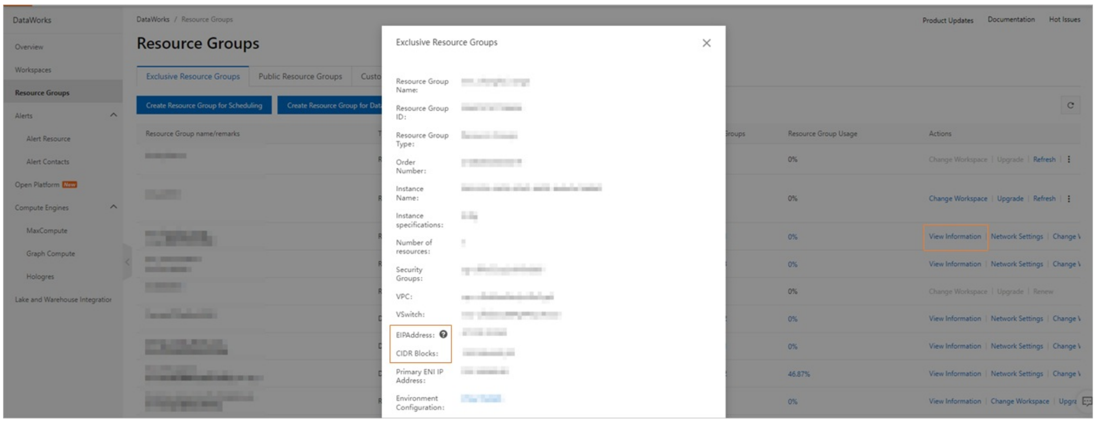
   i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

      a. Log on to the DataWorks console.

      b. In the left-side navigation pane, click **Resource Groups**.

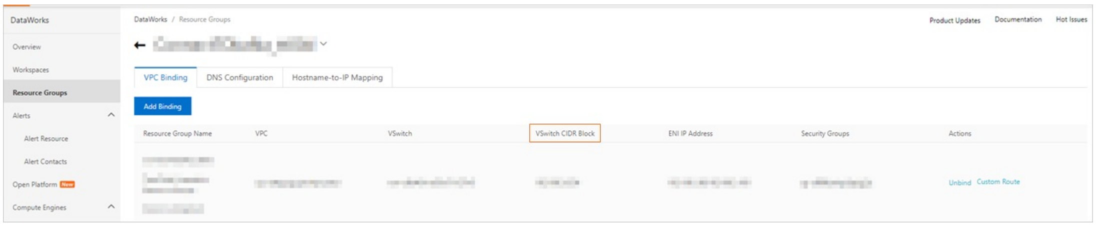
      c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

      d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



      e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

      f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



   ii. Add the elastic IP address (EIP) and CIDR block of the exclusive resource group for data integration to the whitelist of the ApsaraDB for OceanBase cluster. For more information, see Configure a whitelist for an ApsaraDB for OceanBase cluster.

2. Create an account and grant the required permissions to the account.

   You must create an account to log on to the database of the ApsaraDB for OceanBase cluster. You must grant the required permissions to the account. For more information, see Create an account.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add data sources.

3.7.5. Add data sources

Before you configure a real-time sync node to synchronize data from a data source to an AnalyticDB for MySQL data source, you must add both data sources to DataWorks for subsequent source and destination configurations.

Prerequisites

Before you add data sources, make sure that the following operations are performed:

- Prepare data sources: A source data source and a destination data source are created.
- Create and grant permission to an account: An account that is used to access data sources is created.

Precautions

DataWorks provides workspaces in basic mode and standard mode. A workspace in basic mode does not isolate the development environment from the production environment. A workspace in standard mode isolates the development environment from the production environment.

If you use a workspace in standard mode, you must separately add data sources to the development environment and production environment.

Add a source PolarDB for MySQL data source

To add a source PolarDB for MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source PolarDB for MySQL data source, see Add a PolarDB data source.

If the source PolarDB for MySQL data source that you want to add fails the connectivity test, follow the instructions described in What do I do if the PolarDB data source cannot be connected? to handle the exception.

Add a source MySQL data source

To add a source MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source MySQL data source, see Add a MySQL data source.

Add a destination AnalyticDB for MySQL data source

For more information about how to add a destination AnalyticDB for MySQL data source, see Configure an AnalyticDB for MySQL 3.0 connection.

What's next

After you add data sources, you can create and run a data sync node to synchronize data from the source data source to the destination data source.

For more information, see Configure and manage a real-time sync node.

3.7.6. Configure and manage a real-time sync node

After the data sources, network environments, and resource groups are configured, you can create a real-time sync node to synchronize data to an AnalyticDB for MySQL data source. This topic describes how to create a real-time sync node and view the status of the node.

Prerequisites

Before you create a real-time sync node, make sure that the following operations are performed:

- Plan and configure resources
- Configure a source PolarDB data source
- Configure data sources for data synchronization from MySQL
- Configure a source ApsaraDB for OceanBase data source
- Add data sources

Limits

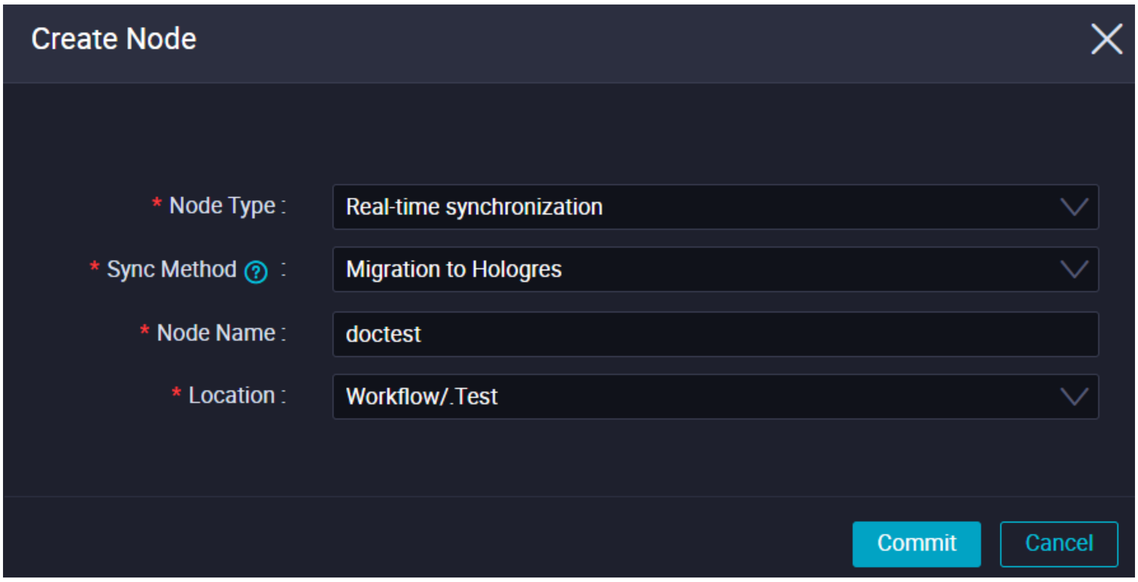
- DataWorks allows you to use only exclusive resource groups for Data Integration to run real-time data sync nodes.
- You can run a real-time sync node to synchronize data to an AnalyticDB for MySQL data source only from a PolarDB, a MySQL, or an ApsaraDB for OceanBase data source.
- A real-time data sync node cannot be used to synchronize data in a table that has no primary key.

Create a real-time sync node

1. Log on to the DataWorks console.

2. In the left-side navigation pane, click **Workspaces**.
3. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
4.
5. This section describes how to create a real-time sync node.
   i. On the DataStudio page, move the pointer over the **+Create** icon and choose **Data Integration > Real-time synchronization**.

      Alternatively, find the workflow in which you want to create a real-time data sync node and right-click the **Data Integration**. From the shortcut menu, choose **Create > Real-time synchronization**.
   ii. Log on to the DataWorks console. In the left-side navigation pane, click Workspaces. On the DataStudio page, create a real-time sync node. In the **Create Node** dialog box, set the parameters as required.



Create Node

* Node Type : Real-time synchronization

* Sync Method : Migration to Hologres

* Node Name : doctest

* Location : Workflow/.Test

Parameter	Description
Node Type	The type of the node. Default value: **Real-time synchronization**.
Sync Method	Set this parameter to **Migration to AnalyticDB MySQL 3.0 in realtime mode**. This setting is used to synchronize data from specified or all tables in a database to an AnalyticDB for MySQL data source.
Node Name	The name of the node. The name must be 1 to 128 characters in length and can contain letters, digits, underscores (_), and periods (.).
Location	The directory in which the real-time sync node is stored.

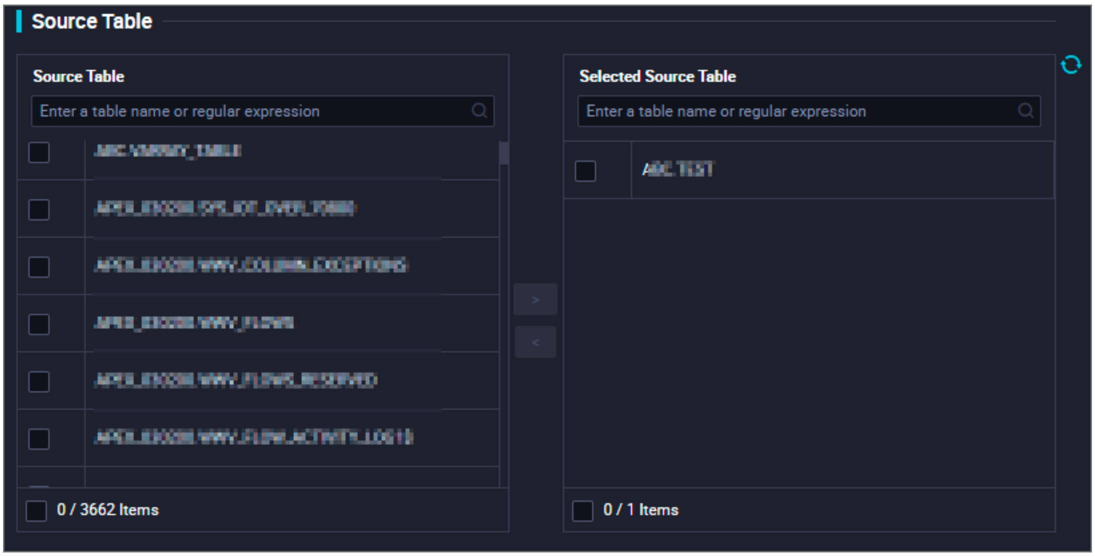
   iii. Click **Commit**. You are navigated to the configuration tab of the real-time data sync node.
6. Select a resource group.
   i. On the right side of the configuration tab, click the **Basic Configuration** tab.
   ii. In the panel that appears, select the resource group that you want to use from the **Resource Group** drop-down list.

> **Note**
>
> DataWorks allows you to use only exclusive resource groups for Data Integration to run real-time data sync nodes.
>
> If no exclusive resource group for Data Integration exists, click **Create Exclusive Resource Group for Data Integration** to create a resource group. For more information, see Overview.

7. Select a source and configure synchronization rules.
    i. In the **Data Source** section, specify the **Type** and **Data source** parameters.

       > ⓘ **Note** You can set the Type parameter only to MySQL, PolarDB, or OceanBase.

    ii. In the **Source Table** section, select the tables whose data you want to synchronize from the **Source Table** list. Then, click the > icon to add the tables to the **Selected Source Table** list.



The Source Table section displays all the tables in the source. You can select all or specific tables.

> **Notice** If a selected table does not have a primary key, the table cannot be synchronized in real time.

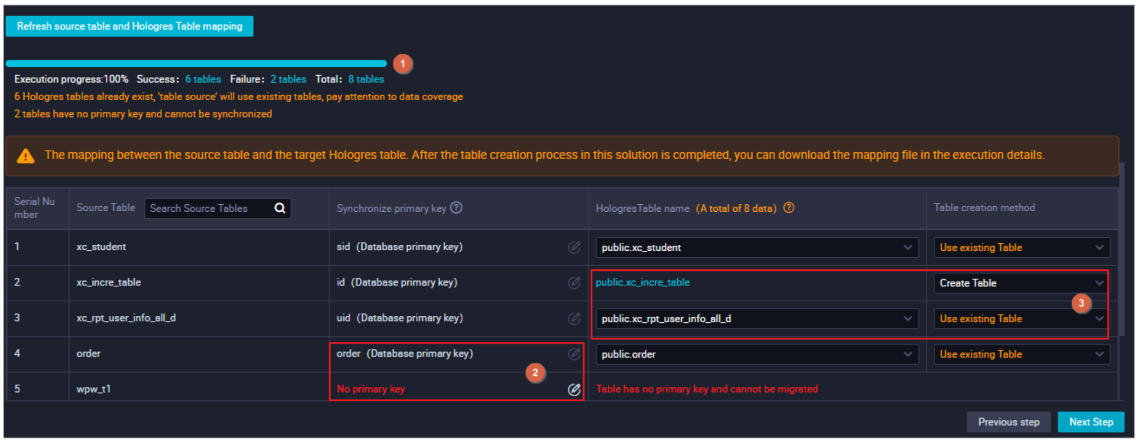
    iii. In the **Mapping Rule for Table Name** section, click **Add rule** to select a rule.

       Supported options are **Conversion Rule for Table Name** and **Rule for Destination Table name**.

       - **Conversion Rule for Table Name**: the rule that is used to convert the names of source tables to those of destination tables.
       - **Rule for destination Table name**: the rule that is used to add a prefix or a suffix to the converted names of destination tables.

    iv. Click **Next Step**.

8. Select a data source as the destination and configure the formats for the destination tables.
    i. In the **Set Destination Table** step, specify the **Target AnalyticDB for MySQL 3.0 data source** parameter.
    ii. Click **Refresh source table and AnalyticDB MySQL 3.0 Table Mapping** to configure the mappings between the source tables to be synchronized and the destination AnalyticDB for MySQL tables.
    iii. View the mapping progress, source tables, and mapped destination tables.



Execution progress:100% Success: 6 tables Failure: 2 tables Total: 8 tables

6 Hologres tables already exist, 'table source' will use existing tables, pay attention to data coverage

2 tables have no primary key and cannot be synchronized

The mapping between the source table and the target Hologres table. After the table creation process in this solution is completed, you can download the mapping file in the execution details.

Serial Number	Source Table Search Source Tables	Synchronize primary key	HologresTable name (A total of 8 data)	Table creation method
1	xc_student	sid (Database primary key)	public.xc_student	Use existing Table
2	xc_incre_table	id (Database primary key)	public.xc_incre_table	Create Table
3	xc_rpt_user_info_all_d	uid (Database primary key)	public.xc_rpt_user_info_all_d	Use existing Table
4	order	order (Database primary key)	public.order	Use existing Table
5	wpw_t1	No primary key	Table has no primary key and cannot be migrated	

Serial number	Description
1	The progress of mapping the source tables to the destination tables. Note The mapping may require a long period of time if data is synchronized from a large number of tables.
2	■ If the tables in the source database contain primary keys, the system removes duplicate data based on the primary keys during the synchronization. ■ If the tables in the source database do not contain primary keys, you can click the ✎ icon to customize primary keys. You can use one field or a combination of several fields as the primary keys of the tables. This way, the system removes duplicate data based on the primary keys during the synchronization. Note A real-time sync node cannot be used to synchronize a table that has no primary key.
3	The method that is used to create a table. Valid values: ■ If you set the **Table creation method** parameter to **Use existing Table**, the names of the automatically created AnalyticDB for MySQL tables are displayed in the **AnalyticDB for MySQL 3.0 Table name** column. You can also select the table name that you want to use from the drop-down list. ■ If you set the **Table creation method** parameter to **Create Table**, the names of the automatically created AnalyticDB for MySQL tables are displayed. To view and modify the SQL statements that are used to create a table, click the name of the table.

iv. Click **Next Step**.

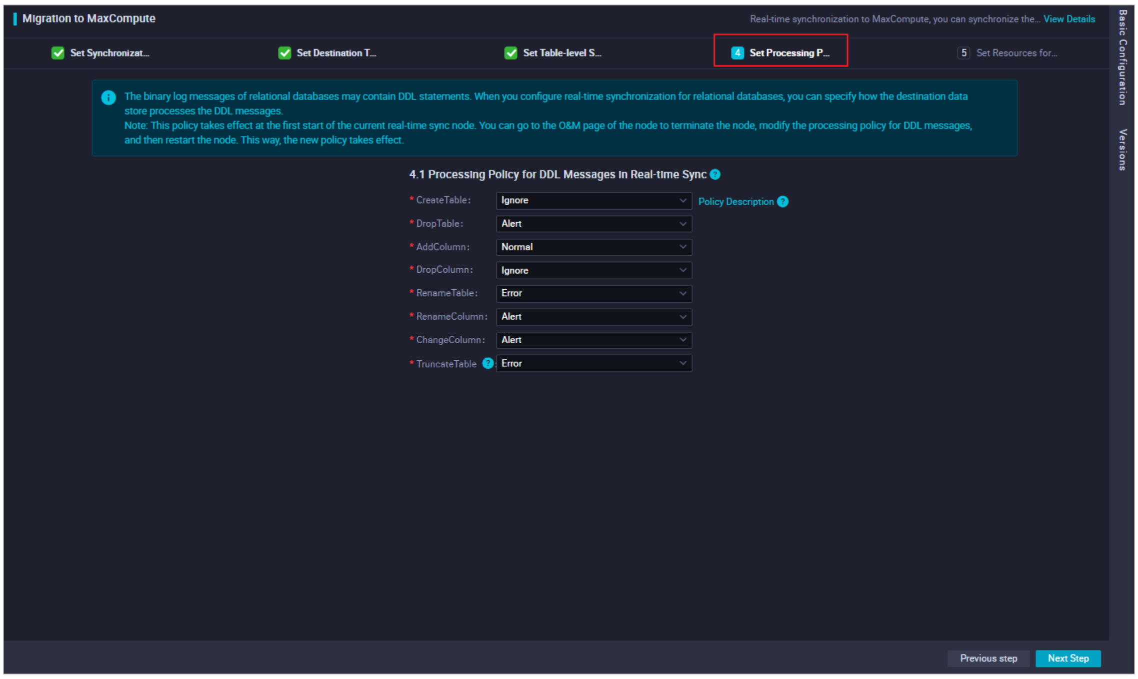
If you set the **Table creation method** parameter to **Create Table**, you must click **Start table building** in the **Create tables automatically** dialog box to create destination AnalyticDB for MySQL tables.

9. Configure rules for processing DDL messages.

   DDL statements exist in the source. Before you synchronize data, you can configure synchronization rules for different DDL statements based on your business requirements.

   > ⓘ **Note** The rules apply when a real-time sync node is run for the first time. If you want to modify the rules in subsequent operations, go to the configuration page of the real-time sync node. For more information, see the "*Start the real-time sync node*" section of this topic.

   i. In the **Set Processing Policy for DDL Messages** step, configure rules to process DDL messages during data synchronization.



Migration to MaxCompute

Real-time synchronization to MaxCompute, you can synchronize the... View Details

✓ Set Synchronizat... ✓ Set Destination T... ✓ Set Table-level S... 4 Set Processing P... 5 Set Resources for...

The binary log messages of relational databases may contain DDL statements. When you configure real-time synchronization for relational databases, you can specify how the destination data store processes the DDL messages.
Note: This policy takes effect at the first start of the current real-time sync node. You can go to the O&M page of the node to terminate the node, modify the processing policy for DDL messages, and then restart the node. This way, the new policy takes effect.

4.1 Processing Policy for DDL Messages in Real-time Sync

- * CreateTable: Ignore
- * DropTable: Alert
- * AddColumn: Normal
- * DropColumn: Ignore
- * RenameTable: Error
- * RenameColumn: Alert
- * ChangeColumn: Alert
- * TruncateTable: Error

The following table describes the processing rules for different DDL messages.

DDL message	Rule
CreateTable	DataWorks processes a DDL message of the related type based on the following rules after it receives the message: ■ **Normal**: sends the message to the destination. Then, the destination processes the message. Each destination may process DDL messages based on its own business logic. If you select Normal for CreateTable, DataWorks only forwards the messages. ■ **Ignore**: ignores the message and does not send it to the destination. ■ **Alert**: ignores the message and records the alert in real-time synchronization logs. In addition, the alert contains information about the reason indicating that a message is ignored because of a running error. ■ **Error**: returns an error when the real-time sync solution is running and terminates the real-time sync solution.
DropTable	
AddColumn	
DropColumn	
RenameTable	
RenameColumn	
ChangeColumn	
TruncateTable	

   ii. Click **Next**.

10. Configure the resources required by the data sync node.

i. In the **Set Resources for Solution Running** step, set the parameters that are described in the following table.

Parameter	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: **15**.
Maximum number of parallel threads allowed to read by destination	The maximum number of parallel threads that the sync node uses to read data from the source table or write data to the destination. Maximum value: 32. Specify an appropriate number based on the resources of the source and the destination.

ii. Click **Complete Configuration**.

Commit and deploy the real-time data sync node

1.

Start the real-time sync node

1. Go to the Operation Center page.

   After you commit and deploy the real-time data sync node, click **Operation Center** in the upper-right corner of the DataStudio page to manage the node on the **Real Time DI** page.

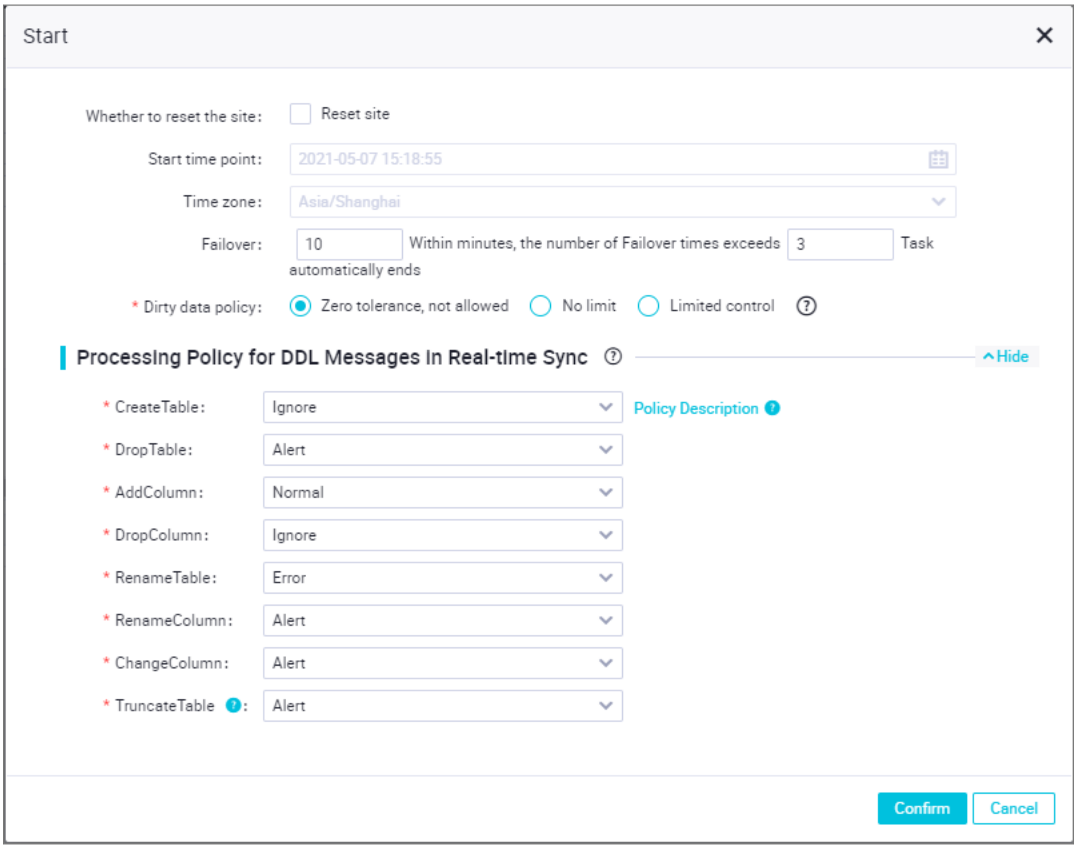
2. View the details of a real-time data sync node.

   On the **Real Time DI** page, find the real-time data sync node that you want to view and click the node name.

3. Start the real-time sync node.

   i. Go back to the previous page and click **Start** in the **Operation** column that corresponds to your desired node.

ii. In the **Start** dialog box, set the parameters as required.



Start

Whether to reset the site: □ Reset site

Start time point: 2021-05-07 15:18:55

Time zone: Asia/Shanghai

Failover: 10 Within minutes, the number of Failover times exceeds 3 Task automatically ends

* Dirty data policy: ● Zero tolerance, not allowed ○ No limit ○ Limited control

Processing Policy for DDL Messages in Real-time Sync

* CreateTable:	Ignore
* DropTable:	Alert
* AddColumn:	Normal
* DropColumn:	Ignore
* RenameTable:	Error
* RenameColumn:	Alert
* ChangeColumn:	Alert
* TruncateTable:	Alert

Parameter	Description
Whether to reset the site	Specifies whether to set the time point for the next startup. If you select the Reset site parameter, the **Start time point** and **Time zone** parameters are required.
Start time point	The date and time for starting the real-time sync node.
Time zone	The time zone in which the real-time sync node is run. You can select a time zone from the **Time zone** drop-down list.
Failover	The maximum number of failovers allowed within the specified time range. **Note** If this parameter is not specified, the system automatically stops the node if the number of failovers exceeds 100 within 5 minutes. This avoids excessive resource consumption caused by the frequent starting of the node.
Dirty data policy	■ **Zero tolerance, not allowed**: The real-time sync node is automatically stopped if the node contains dirty data. ■ **No limit**: The real-time sync node can normally run regardless of whether the node contains dirty data. ■ **Limited control**: The real-time sync node is automatically stopped if the amount of dirty data contained in the node exceeds a specified value.
Processing Policy for DDL Messages in Real-time Sync	You can modify the configured rules that are used to process DDL message based on your business requirements. For more information, see Step 10 in this topic.

Manage the real-time data sync node

- Stop a real-time data sync node that is running.

  Find the real-time data sync node that you want to stop and click **Stop** in the Operation column. In the message that appears, click **Stop**.

- Undeploy a real-time data sync node that is not running.

  Find the real-time data sync node that you want to undeploy and click **Undeploy** in the Operation column. In the message that appears, click **Undeploy**.

- View the alert information of a real-time data sync node.

  Find the real-time data sync node that you want to view and click **Alert settings** in the Actions column. In the **Alert settings** dialog box, view the alert events and alert rules.

- Configure alert rules for a real-time data sync node.

  i. Find the real-time data sync node for which you want to configure alert rules and click **Configure Alert Rule** in the lower part of the **Real Time DI** page.

  ii. In the **New rule** dialog box, set the parameters that are described in the following table.

Parameter	Description
Name	The name of the alert rule.
Description	The description of the alert rule.
Indicators	The metric for which an alert is reported. Valid values: ■ **Status** ■ **Business delay** ■ **Failover** ■ **Dirty Data** ■ **Not Supported by DDL Statement**
Threshold	The threshold for reporting an alert. Specify the **WARNING In** and **CRITICAL In** parameters. The default values of the parameters are 5 minutes.
Alarm interval	The interval at which an alert is reported. The default value is 5 minutes.
WARNING	The method that is used to send alert notifications. You can specify one or more methods. Valid values: **Mail**, **SMS**, and **DingTalk**. **Note** Only Singapore, Malaysia(Kuala Limpur), and Germany(Frankfurt) support the SMS reminding method. To use the SMS reminding method in other regions, submit a ticket to contact DataWorks technical support.
CRITICAL	
Receiver (Non-DingTalk)	The recipient of alert notifications.

  iii. Click **Confirm**.

- Modifies alert rules for real-time data sync nodes at a time.

  i. Select one or more real-time data sync nodes for which you want to modify alert rules and click **Operation alarm** in the lower part of the **Real Time DI** page.

  ii. In the **Operation alarm** dialog box, modify the values of the **Type** and **Indicators** parameters.

  iii. Click **Confirm**.

3.7.7. FAQ

The topic provides answers to frequently asked questions when you fail to synchronize data to an AnalyticDB for MySQL V3.0 data source in real time.

- What do I do if a PolarDB data source cannot be connected?

- What do I do if a MySQL data source cannot be connected?
- The system displays the following error message for a real-time sync node: "com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX." What do I do?
- The system displays the following error message for a real-time sync node: "com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation." What do I do?
- The system displays the following error message for a real-time sync node: "com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first." What do I do?

What do I do if a PolarDB data source cannot be connected?

- Problem description: The connectivity test fails when I add a PolarDB data source.
- Solution: Set Data source type to Connection string mode and check the whitelist configuration of the PolarDB cluster and the virtual private cloud (VPC) configuration of your exclusive resource group.

What do I do if a MySQL data source cannot be connected?

- Problem description: The connectivity test fails when I add a MySQL data source.
- Solution: Set Data source type to Connection string mode and check the whitelist configuration of the PolarDB cluster and the virtual private cloud (VPC) configuration of your exclusive resource group.

The system displays the following error message for a real-time sync node: "com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX." What do I do?

- Problem description: The real-time synchronization node fails to run, and the system displays the error message " `com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX` ."
- Cause: The binary logging feature is disabled for the PolarDB data source.
- Solution: Enable the binary logging feature for the PolarDB data source. For more information, see Configure a data source (PolarDB). In addition, you must change one or more data records and change the start time for running the real-time sync node to the current time.

The system displays the following error message for a real-time sync node: "com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation." What do I do?

- Problem description: The real-time synchronization node fails to run, and the system displays the error message " `com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation` ."
- Cause: The account used to synchronize data is not authorized to access the PolarDB data source, or the PolarDB database connected is not deployed on the PolarDB Writer node.
- Solution: Authorize the account to access the PolarDB data source. For more information, see Configure a data source (PolarDB). You can also check whether the PolarDB database connected is deployed on the PolarDB Writer node. When a real-time sync node is run, the system cannot capture data from the PolarDB Reader nodes of the PolarDB cluster.

The system displays the following error message for a real-time sync node: "com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first." What do I do?

- Problem description: The real-time synchronization node fails to run, and the system displays the error message " `com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first` ."
- Cause: The loose_polar_log_bin parameter is not specified for the PolarDB data source.
- Solution: Specify the loose_polar_log_bin parameter. For more information, see Configure a data source (PolarDB).

3.8. Synchronize all data in a database to DataHub

3.8.1. Plan and configure resources

When you use DataWorks to synchronize data, you can use only exclusive resource groups for Data Integration to run real-time sync nodes. This topic describes the resources and configurations required to run real-time sync nodes.

Context

- Resource planning and preparation

  Before you use a data synchronization node to synchronize data, you must purchase an exclusive resource group for data integration and add the resource group to DataWorks for subsequent use.

  For more information about exclusive resource groups for Data Integration, see Exclusive resources for Data Integration.

- Network connections

  An exclusive resource group for Data Integration is essentially a group of Elastic Compute Service (ECS) instances. After you purchase and create such an exclusive resource group, it is isolated from other services. You must associate the resource group with a virtual private cloud (VPC) to ensure network connectivity between the resource group and data sources during subsequent data synchronization.

What's next

After you plan and configure resources, you can configure data sources. You must configure network connections for the data sources and permissions to access the data sources. This facilitates the creation of a real-time sync node. You can synchronize data only from PolarDB, ApsaraDB for OceanBase, MySQL, or Oracle to DataHub. For more information about how to configure a data source, see Configure data sources for data synchronization from PolarDB, Configure a data source (ApsaraDB for OceanBase), Configure data sources for data synchronization from MySQL, and Configure data sources for data synchronization from Oracle.

3.8.2. Configure data sources for data synchronization from PolarDB

Before you use DataWorks to synchronize data from a PolarDB data source to DataHub, you can refer to the operations in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions.

Prerequisites

Before you configure the data sources, make sure that the following operations are performed:

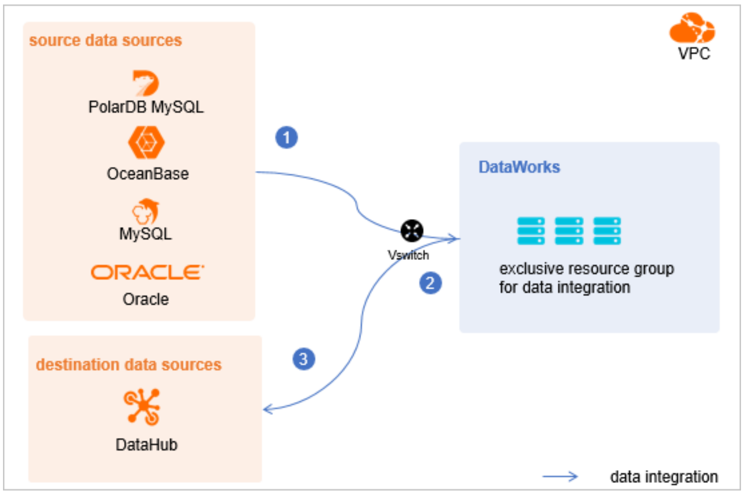
- Prepare data sources: A source PolarDB for MySQL cluster and a destination DataHub project are prepared. In this topic, a PolarDB for MySQL cluster is used as the source.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

  If the source and destination data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source is a PolarDB for MySQL cluster, you must enable the binary logging feature for the cluster. PolarDB for MySQL is fully compatible with MySQL and uses high-level physical logs to replace binary logs. To facilitate the integration between PolarDB and the MySQL ecosystem, you can enable the binary logging feature for PolarDB clusters.

Limits

- Only PolarDB for MySQL clusters can be used as sources in data synchronization solutions. Other types of PolarDB data sources are not supported. In this topic, PolarDB indicates PolarDB for MySQL data sources.
- Only data stored on the primary node of a PolarDB for MySQL cluster can be synchronized.

Procedure

1. Configure a whitelist for the PolarDB for MySQL cluster.

   To add the CIDR block of the VPC where the exclusive resource group for Data Integration resides to a whitelist of the PolarDB for MySQL cluster, perform the following steps:

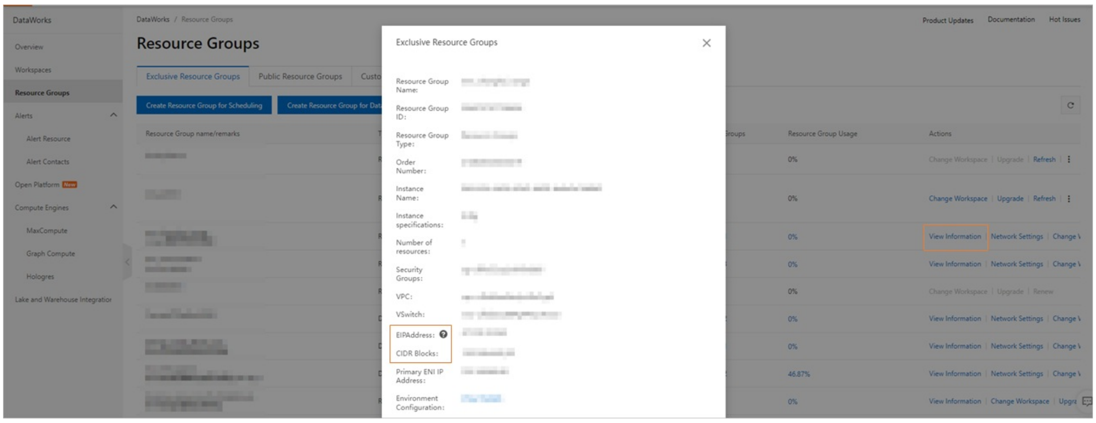
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

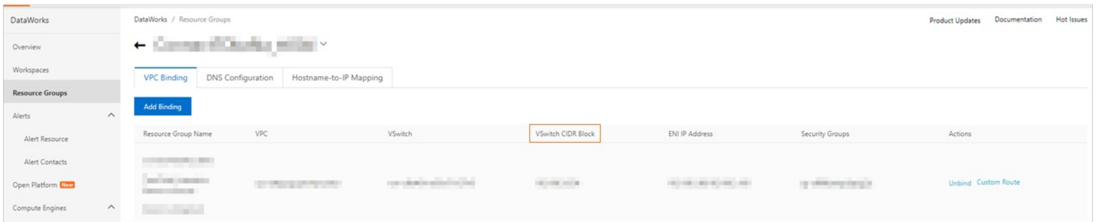
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.

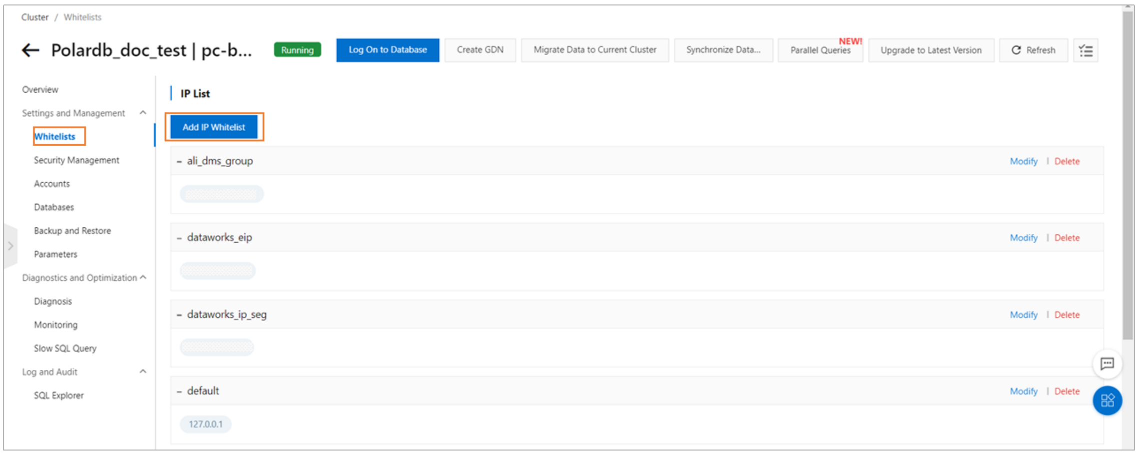


e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR blocks recorded in the preceding steps to the whitelist of the PolarDB for MySQL cluster.



For more information, see Configure an IP whitelist.

2. Create an account and grant the required permissions to the account.

You must create an account to log on to the database of the PolarDB for MySQL cluster. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

i. Create an account.

For more information, see Create a database account.

ii. Grant the required permissions to the account.

You can run the following command to grant the required permissions to the account, or you can directly assign the `SUPER` role to the account.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Account for data synchronization';
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%';
```

3. Enable the binary logging feature for the PolarDB for MySQL cluster.

   For more information, see Enable binary logging.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add data sources.

3.8.3. Configure data sources for data synchronization from MySQL

Before you synchronize data from MySQL to DataHub in real time, you can refer to the operations described in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions. You must configure a source MySQL data source and a destination DataHub data source.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

- Prepare data sources: A MySQL database and a DataHub project are prepared.
- Plan and prepare resources: An exclusive resource group for data integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.
- Execute the following statement to check the version of your MySQL database. You can use a `MySQL 5.x` or `MySQL 8.x` database.

  ```
  select version();
  ```

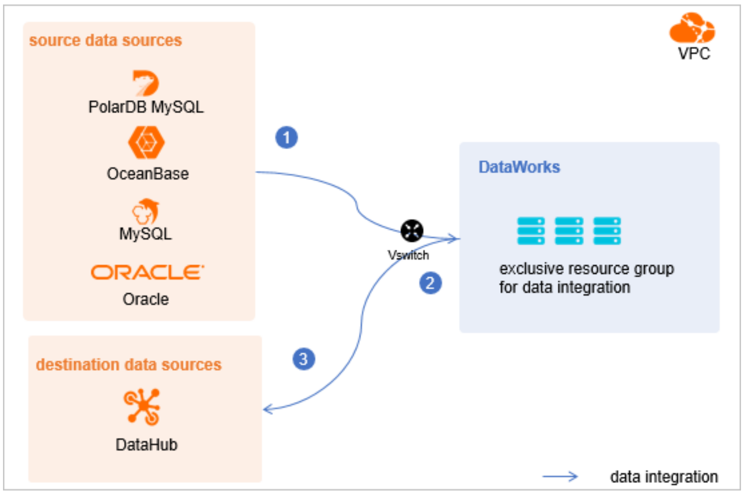
  Note Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.X` or `V8.X`. PolarDB for MySQL is not supported. If the MySQL version of your ApsaraDB RDS for MySQL database is not `V5.X` or `V8.X`, use an ApsaraDB RDS for MySQL database whose MySQL version is `V5.X` or `V8.X`. Otherwise, the data synchronization node fails to run.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for data sources

If the data sources and exclusive resource group for data integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for data integration to the whitelists of the data sources. This ensures that the exclusive resource group for data integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source data source is a MySQL database, you must enable the binary logging feature. Binary logs record changes to all table schemas and modifications on table data. You can execute statements such as CREATE and ALTER to perform operations on table schemas. You can execute statements such as INSERT, UPDATE, and DELETE to perform operations on table data. You can use binary logs to view the change history of the database, back up incremental data and restore data in the database, and replicate data from the primary database to secondary databases.

  Formats of binary logs:

  - Statement: SQL statement-based replication. Binary logs in this format record the SQL statements that are executed to modify data entries.
  - Row: row-based replication. Binary logs in this format record only modification details about data entries in rows.
  - Mixed: replication in mixed mode. This mode combines the statement and row formats. In most cases, binary logs in the statement format are used to record the SQL statements that are executed to modify data entries, such as functions. If data replication from the primary database to secondary databases cannot be implemented by using binary logs in this format, switch to the row format. MySQL determines which format to use based on each SQL statement that is executed.

Limits

Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.x` or `V8.x`. PolarDB for MySQL is not supported.

Procedure

1. Configure a whitelist for the MySQL database.

   Add the CIDR block of the VPC in which the exclusive resource group resides to the whitelist of the MySQL database.

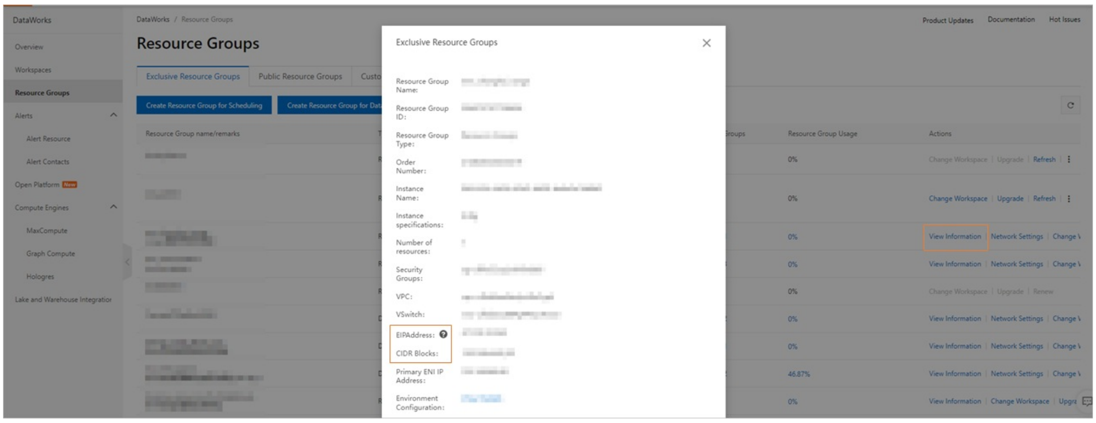
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

   a. Log on to the DataWorks console.

   b. In the left-side navigation pane, click **Resource Groups**.

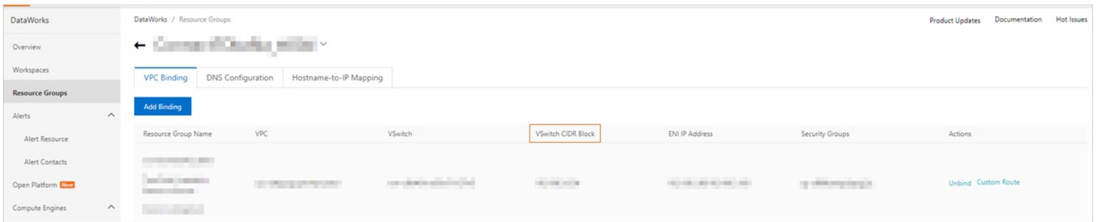
   c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

   d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



   e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

   f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR block of the exclusive resource group to the whitelist of the MySQL database.

2. Create an account and grant permissions to the account.

   You must create an account to log on to the MySQL database. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

   i. Create an account.

   For more information, see Create an account to access a MySQL database.

ii. Grant permissions to the account.

You can run the following command to grant permissions to the account. Alternatively, you can grant the `SUPER` permission to the account. Replace `Account for data synchronization` with the created account when you execute the specific statement.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Password'; // Create an account th
at is used to synchronize data and set a password so that you can use the account and password to acces
s the database from a host. % indicates a host.
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%'; /
/ Grant the SELECT, REPLICATION SLAVE, and REPLICATION CLIENT permissions to the account.
```

`*.*` indicates that the synchronization account is granted the preceding permissions on all tables in all databases. You can also grant the preceding permissions on specified tables in the specified database to the synchronization account. For example, to grant the account the preceding permissions on the *user* table in the *test* database, execute the following statement: `GRANT SELECT, REPLICATION CLIENT ON test.user TO 'Account for data synchronization'@'%';` .

> **Note** The `REPLICATION SLAVE` permission is a global permission. You cannot grant this permission on specified tables in the specified database to the synchronization account.

3. Enable the binary logging feature for the MySQL database.

Perform the following steps to check whether the binary logging feature is enabled and to query the format of binary logs:

- Execute the following statement to check whether the binary logging feature is enabled:

```
show variables like "log_bin";
```

If *ON* is displayed in the returned result, the binary logging feature is enabled.

- If you use a secondary database to synchronize data, execute the following statement to check whether the binary logging feature is enabled:

```
show variables like "log_slave_updates";
```

If *ON* is displayed in the returned result, the binary logging feature is enabled for the secondary database.

If the returned result is different from the preceding result, follow the instructions described in the *MySQL documentation* to enable the binary logging feature.

Execute the following statement to view the format of binary logs:

```
show variables like "binlog_format";
```

Returned result:

- *ROW*: The format of binary logs is row.
- *STATEMENT*: The format of binary logs is statement.
- *MIXED*: The format of binary logs is mixed.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add data sources.

3.8.4. Configure a data source (ApsaraDB for OceanBase)

Before you use DataWorks to synchronize data from ApsaraDB for OceanBase to DataHub, you must refer to the operations in this topic to prepare the configurations, such as network environments, whitelists, and permissions, for both the source and destination.

Prerequisites

Before you configure a data source, make sure that the following operations are performed:

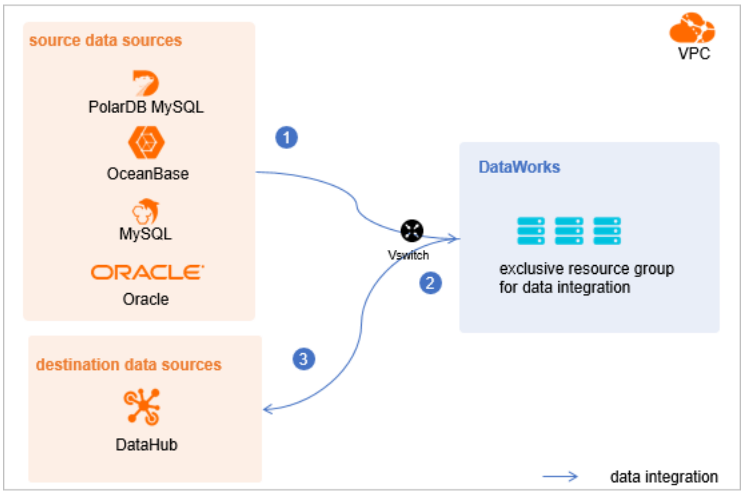
- Prepare data sources: An ApsaraDB for OceanBase cluster and a DataHub project are prepared.
- Plan and prepare resources: An exclusive resource group for data integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
    - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
    - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and exclusive resource groups for data integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for data sources

  If the data sources and exclusive resource group for data integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for data integration to the whitelists of the data sources. This ensures that the exclusive resource group for data integration can be used to access the data sources.

- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

Limits

ApsaraDB for OceabBase is a distributed relational database that can integrate data distributed in multiple physical databases into a unified logical database. However, you can synchronize data of only one physical ApsaraDB for OceanBase database to DataHub in real time.

Procedure

1. Configure a whitelist for the ApsaraDB for OceanBase cluster.

   Add the CIDR block of the VPC where the exclusive resource group resides to the whitelist of the ApsaraDB for OceanBase cluster.

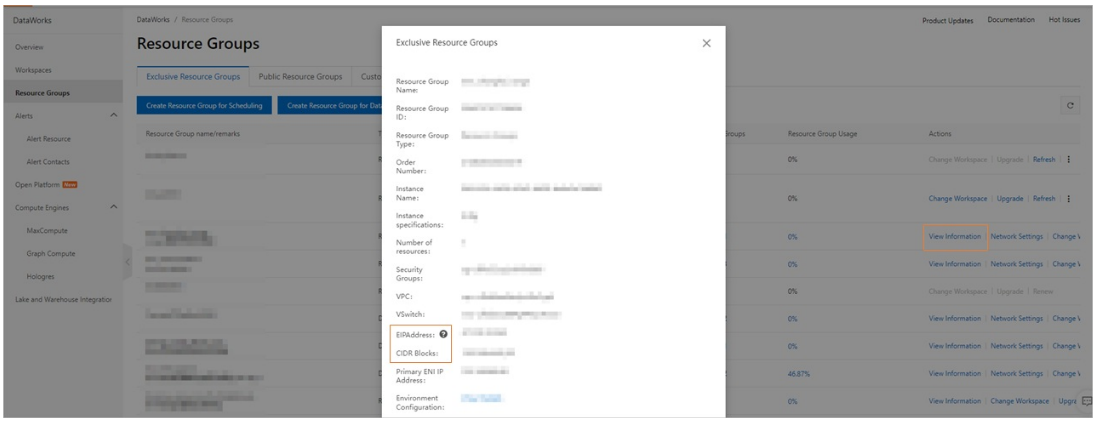
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

  a. Log on to the DataWorks console.

  b. In the left-side navigation pane, click **Resource Groups**.

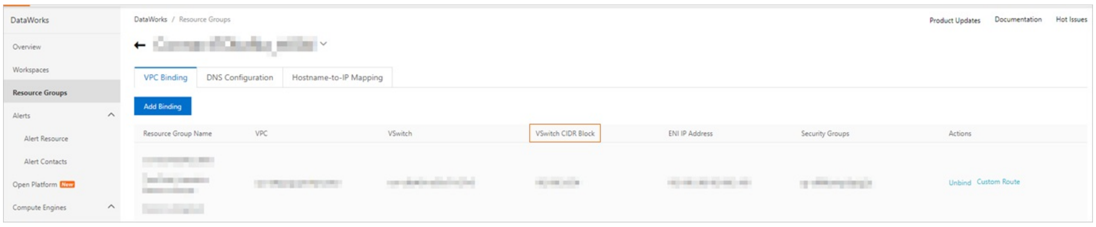
  c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

  d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



  e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

  f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the elastic IP address (EIP) and CIDR block of the exclusive resource group for data integration to the whitelist of the ApsaraDB for OceanBase cluster. For more information, see *Configure a whitelist for an ApsaraDB for OceanBase cluster* of OceanBase.

2. Create an account and grant the required permissions to the account.

   You must create an account to log on to the database of the ApsaraDB for OceanBase cluster. You must grant the required permissions on the database to the account. For more information, see *Create an account* of OceanBase.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add data sources.

3.8.5. Configure data sources for data synchronization from Oracle

Before you synchronize data from Oracle to DataHub in real time, you can refer to the operations described in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions. You must configure a source Oracle data source and a destination DataHub data source.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

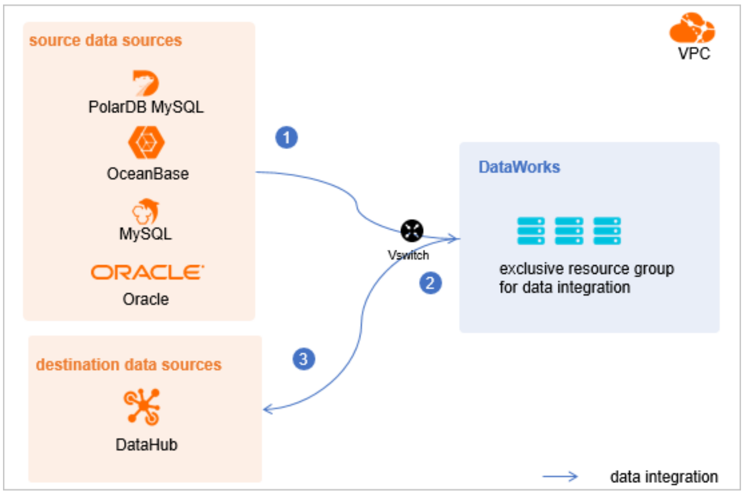
- Prepare data sources: An Oracle database and a DataHub project are prepared.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
    - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
    - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources. The Oracle database must contain only the database versions, character encoding formats, and data types that are supported by Data Integration.

- Configure whitelists for the data sources

  If the source and destination data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

-
-
-
-

Limits

- Only DataWorks workspaces in the China (Shenzhen) region support real-time synchronization of all data in a database from Oracle to DataHub. If you want to synchronize data from Oracle to DataHub in workspaces of other regions, submit a ticket.
-
-
-
-

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add data sources.

3.8.6. Add data sources

Before you configure a real-time data sync node to synchronize data from a data source to the DataHub data source, you must add both data sources to DataWorks for subsequent source and destination configurations.

Prerequisites

Before you add data sources, make sure that the following operations are performed:

- Prepare data sources: A source data source and a destination data source are created.
- Create and grant permission to an account: An account that is used to access data sources is created.

Precautions

DataWorks provides workspaces in basic mode and standard mode. A workspace in basic mode does not isolate the development environment from the production environment. A workspace in standard mode isolates the development environment from the production environment.

If you use a workspace in standard mode, you must separately add data sources to the development environment and production environment.

Add a source PolarDB for MySQL data source

To add a source PolarDB for MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source PolarDB for MySQL data source, see Add a PolarDB data source.

If the source PolarDB for MySQL data source that you want to add fails the connectivity test, follow the instructions described in What do I do if the PolarDB data source cannot be connected? to handle the exception.

Add ApsaraDB for OceanBase as the source

To add ApsaraDB for OceanBase as the source, you must configure information such as the network connection type, and the access account and password as planned. For more information about how to add ApsaraDB for OceanBase as the source, see Add an ApsaraDB for OceanBase data source.

Add a source MySQL data source

To add a source MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source MySQL data source, see Add a MySQL data source.

Add DataHub as the destination

For more information, see Add a DataHub data source.

Add a source Oracle data source

To add a source Oracle data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source Oracle data source, see Add an Oracle data source.

What's next

After you add data sources, you can create and run a data sync node to synchronize data from the source data source to the destination data source.

For more information, see Configure and manage a real-time data synchronization node.

3.8.7. Configure and manage a real-time data synchronization node

After you prepare data sources, network environments, and resources, you can create a real-time data synchronization node to synchronize data to DataHub. This topic describes how to create a real-time data synchronization node and view the status of the node.

Prerequisites

Before you create a real-time data synchronization node, make sure that the following operations are performed:

- Plan and configure resources
- Configure data sources for data synchronization from PolarDB
- Configure a data source (ApsaraDB for OceanBase)
- Configure data sources for data synchronization from MySQL
- Configure data sources for data synchronization from Oracle
- Add data sources

Limits

- DataWorks allows you to use only exclusive resource groups for Data Integration to run real-time data sync nodes.
- You can use a real-time synchronization node to synchronize data only from a PolarDB, MySQL, or Oracle data source to DataHub.
-

Usage notes

For information about support of different topic types for synchronization of data changes caused by operations on a source table, sharding strategies for different topic types, data formats, and sample messages.

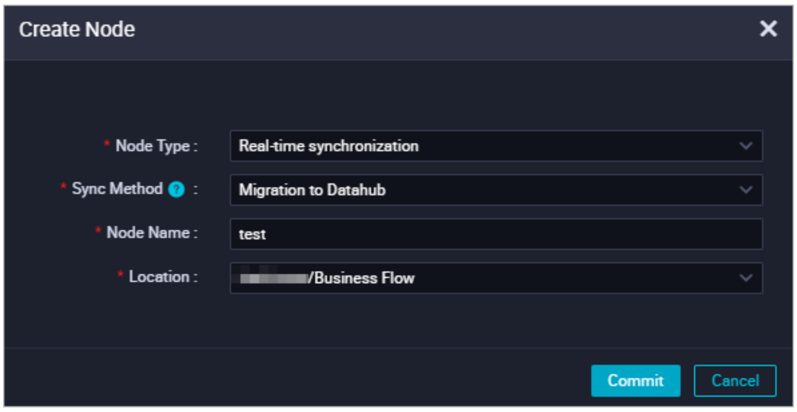
Create a real-time data synchronization node

1. Log on to the DataWorks console.
2. In the left-side navigation pane, click **Workspaces**.
3. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
4.
5. Create a real-time data synchronization node.

   i. On the DataStudio page, move the pointer over the +Create icon and choose **Data Integration > Real-time synchronization**.

      Alternatively, find the workflow in which you want to create a real-time data sync node and right-click the **Data Integration**. From the shortcut menu, choose **Create > Real-time synchronization**.

ii. In the **Create Node** dialog box, configure the parameters.



Create Node

* Node Type : Real-time synchronization

* Sync Method : Migration to Datahub

* Node Name : test

* Location : [redacted]/Business Flow

Parameter	Description
Node Type	The type of the node. Default value: **Real-time synchronization**.
Sync Method	Set the value to **Migration to Datahub**. In this case, some or all tables in your desired data source are migrated to DataHub.
Name	The name of the node. The name cannot exceed 128 characters in length and can contain only letters, digits, underscores (_), and periods (.).
Path	The directory in which the real-time data synchronization node is stored.

iii. Click **Commit**. You are navigated to the configuration tab of the real-time data sync node.

6. Select a resource group.

   i. On the right side of the configuration tab, click the **Basic Configuration** tab.

   ii. In the panel that appears, select the resource group that you want to use from the **Resource Group** drop-down list.

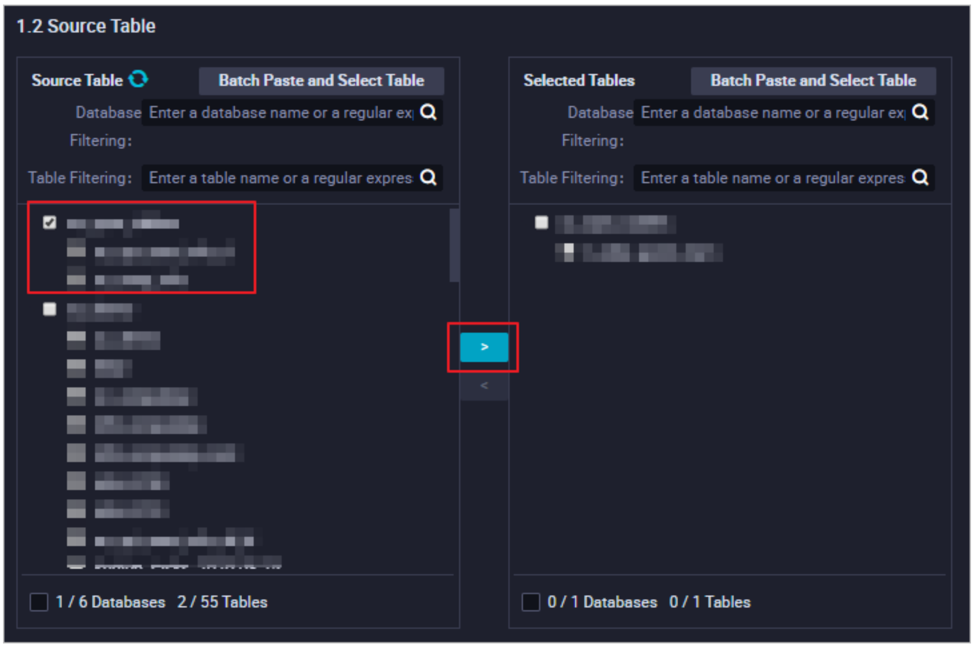
   > Note
   >
   > DataWorks allows you to use only exclusive resource groups for Data Integration to run real-time data sync nodes.
   >
   > If no exclusive resource group for Data Integration exists, click **Create Exclusive Resource Group for Data Integration** to create a resource group. For more information, see Overview.

7. Select a data source as the source and configure synchronization rules.

   i. In the **Data Source** section, configure the **Type** and **Data source** parameters.

   > Note You can set the Type parameter only to MySQL, Oracle, or PolarDB.

ii. In the **Source Table** section, select the tables whose data you want to synchronize from the **Source Table** list. Then, click the > icon to add the tables to the **Selected Source Table** list.



1.2 Source Table

Source Table
Batch Paste and Select Table
Database Filtering:
Enter a database name or a regular ex
Table Filtering:
Enter a table name or a regular expres
1 / 6 Databases 2 / 55 Tables
>
<
Selected Tables
Batch Paste and Select Table
Database Filtering:
Enter a database name or a regular ex
Table Filtering:
Enter a table name or a regular expres
0 / 1 Databases 0 / 1 Tables

The Source Table list displays all tables in the source. You can select all or specific tables.

iii. In the **Conversion Rule for Table Name** section, click **Add Rule** to select a rule.

Supported options include **Conversion Rule for Table Name** and **Rule for Destination Topic**.

- **Conversion Rule for Table Name**: the rule for converting the names of source tables to those of destination topics.
- **Rule for Destination Topic**: the rule for adding prefixes and suffixes to destination topics.

iv. Click **Next**.

8. Select a data source as the destination and configure formats for the destination topics.

i. In the **Set Destination Topic** step, set the **Destination**, **Datahub write mode**, and **Sharding Strategy** parameters.

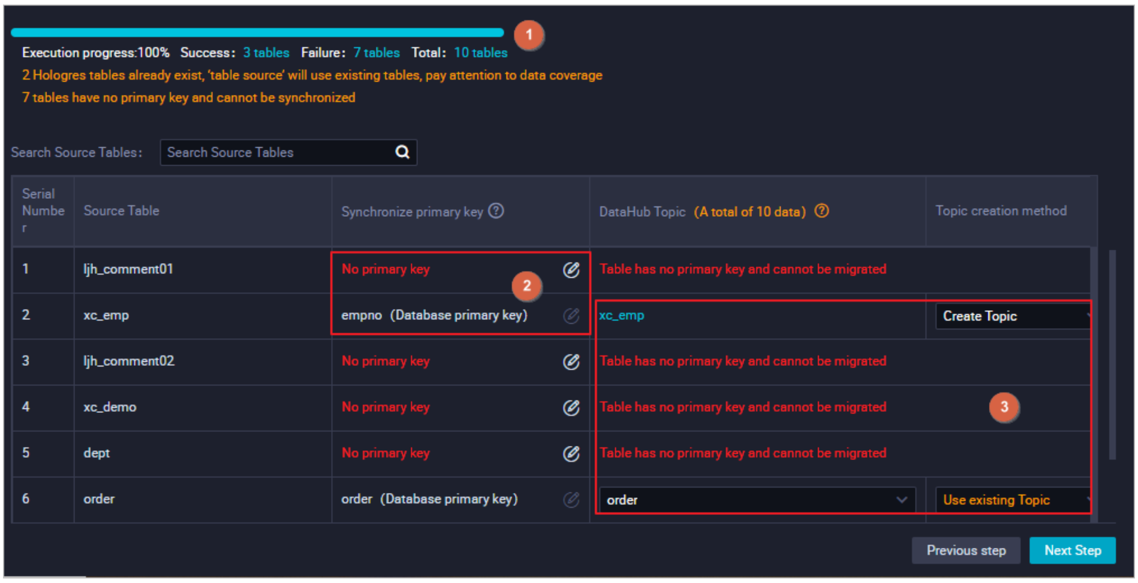
If you want to synchronize source tables that do not have primary keys, you can select **Source tables without primary keys can be synchronized**.

ii. (Optional)Add fields to the destination tables

If you want to add fields to all the tables to be synchronized, click **New field** in the **Fields In Destination Table** section.

iii. Click **Refresh source table and DataHub Topic mapping** to configure the mappings between the source tables and destination DataHub topics.

iv. View the mapping progress, source tables, and mapped destination topics.



Execution progress:100% Success: 3 tables Failure: 7 tables Total: 10 tables

2 Hologres tables already exist, 'table source' will use existing tables, pay attention to data coverage

7 tables have no primary key and cannot be synchronized

Search Source Tables: Search Source Tables

Serial Number	Source Table	Synchronize primary key	DataHub Topic (A total of 10 data)	Topic creation method
1	ljh_comment01	No primary key	Table has no primary key and cannot be migrated	
2	xc_emp	empno (Database primary key)	xc_emp	Create Topic
3	ljh_comment02	No primary key	Table has no primary key and cannot be migrated	
4	xc_demo	No primary key	Table has no primary key and cannot be migrated	
5	dept	No primary key	Table has no primary key and cannot be migrated	
6	order	order (Database primary key)	order	Use existing Topic

No.	Description
1	The progress of mapping the source tables to the destination tables. Note The mapping may require a long period of time if data is synchronized from a large number of tables.
2	- If the tables in the source database have primary keys, the system removes duplicate data based on the primary keys during data synchronization. - If the tables in the source database do not have primary keys, the following situations occur: - If you select **Source tables without primary keys can be synchronized.** in the **Set Destination Topic** step, source tables without primary keys can be synchronized. You can also click the ✎ icon to customize primary keys for the source tables. You can use one field or a combination of several fields in the source tables as primary keys. This way, the system removes duplicate data based on the primary keys during data synchronization. - If you do not select **Source tables without primary keys can be synchronized.** in the **Set Destination Topic** step, errors occur when you synchronize source tables without primary keys. In this case, you must delete these tables or customize primary keys for the tables before the data synchronization can continue.

No.	Description
3	The method that is used to create a destination topic. The message that appears in the DataHub Topic column varies based on the method that you select from the drop-down list in the Topic creation method column. ■ If you select **Create Topic** from the drop-down list in the **Topic creation method** column, the name of the topic that is automatically created appears in the DataHub Topic column. You can click the topic name to modify the topic information. ■ If you select **Use Existing Topic** from the drop-down list in the **Topic creation method** column, you must select the desired topic from the drop-down list in the DataHub Topic column.

v. Click **Next Step**.

If you select **Create Topic** from the drop-down list in the **Topic creation method** column, you must click **Start table building** in the **Create Table** dialog box to create DataHub topics.

9. Configure the resources required by the data sync node.

i. In the **Set Resources for Solution Running** step, set the parameters that are described in the following table.

Parameter	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: **15**.
Maximum number of parallel threads allowed to read by destination	The maximum number of parallel threads that the sync node uses to read data from the source table or write data to the destination. Maximum value: 32. Specify an appropriate number based on the resources of the source and the destination.

ii. Click **Complete Configuration**.

Commit and deploy the real-time data sync node

1.

Run the real-time data synchronization node

1. Go to the Operation Center page.

   After you commit and deploy the real-time data sync node, click **Operation Center** in the upper-right corner of the DataStudio page to manage the node on the **Real Time DI** page.

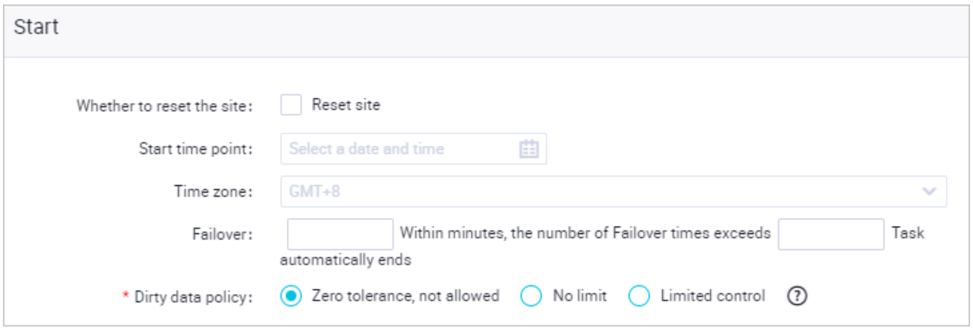
2. View the details of a real-time data sync node.

   On the **Real Time DI** page, find the real-time data sync node that you want to view and click the node name.

3. Run the real-time data synchronization node.

   i. Go to the Real Time DI page of Operation Center, find the real-time data synchronization node that you created and click **Start** in the **Operation** column.

ii. In the **Start** dialog box, configure the parameters.



Start
Whether to reset the site: □ Reset site
Start time point: Select a date and time
Time zone: GMT+8
Failover: Within minutes, the number of Failover times exceeds Task automatically ends
* Dirty data policy: ● Zero tolerance, not allowed ○ No limit ○ Limited control ?

Parameter	Description
Whether to Set Start Offset	Specifies whether to set the point in time for the next startup. If you select Reset site, the **Start time point** and **Time zone** parameters are required.
Time	The date and time for starting the real-time data synchronization node.
Time zone	The time zone in which the real-time data synchronization node is run. You can select a time zone from the **Time zone** drop-down list.
Failover	The maximum number of failovers allowed within the specified time range. **Note** If you do not configure this parameter, the system automatically stops the node if the number of failovers exceeds 100 within 5 minutes. This prevents excessive resource consumption caused by the frequent starting of the node.
Dirty data policy	■ **Zero tolerance, not allowed**: The real-time data synchronization node is automatically stopped if dirty data is generated during data synchronization. ■ **No limit**: The real-time data synchronization node can normally run regardless of whether dirty data is generated during data synchronization. ■ **Limited control**: The real-time data synchronization node is automatically stopped if the amount of dirty data that is generated during data synchronization exceeds a specified value.

iii. Click **Confirm**.

Manage the real-time data sync node

- Stop a real-time data sync node that is running.

  Find the real-time data sync node that you want to stop and click **Stop** in the Operation column. In the message that appears, click **Stop**.

- Undeploy a real-time data sync node that is not running.

  Find the real-time data sync node that you want to undeploy and click **Undeploy** in the Operation column. In the message that appears, click **Undeploy**.

- View the alert information of a real-time data sync node.

  Find the real-time data sync node that you want to view and click **Alert settings** in the Actions column. In the **Alert settings** dialog box, view the alert events and alert rules.

- Configure alert rules for a real-time data sync node.

  i. Find the real-time data sync node for which you want to configure alert rules and click **Configure Alert Rule** in the lower part of the **Real Time DI** page.

  ii. In the **New rule** dialog box, set the parameters that are described in the following table.

Parameter	Description
Name	The name of the alert rule.
Description	The description of the alert rule.
Indicators	The metric for which an alert is reported. Valid values: ■ **Status** ■ **Business delay** ■ **Failover** ■ **Dirty Data** ■ **Not Supported by DDL Statement**
Threshold	The threshold for reporting an alert. Specify the **WARNING In** and **CRITICAL In** parameters. The default values of the parameters are 5 minutes.
Alarm interval	The interval at which an alert is reported. The default value is 5 minutes.
WARNING	The method that is used to send alert notifications. You can specify one or more methods. Valid values: **Mail**, **SMS**, and **DingTalk**. **Note** Only Singapore, Malaysia(Kuala Limpur), and Germany(Frankfurt) support the SMS reminding method. To use the SMS reminding method in other regions, submit a ticket to contact DataWorks technical support.
CRITICAL	
Receiver (Non-DingTalk)	The recipient of alert notifications.

iii. Click **Confirm**.

- Modifies alert rules for real-time data sync nodes at a time.

  i. Select one or more real-time data sync nodes for which you want to modify alert rules and click **Operation alarm** in the lower part of the **Real Time DI** page.

  ii. In the **Operation alarm** dialog box, modify the values of the **Type** and **Indicators** parameters.

  iii. Click **Confirm**.

3.9. Create, configure, commit, and manage real-time sync nodes

DataWorks supports real-time synchronization. This topic describes how to create, configure, commit, and manage real-time sync nodes.

Prerequisites

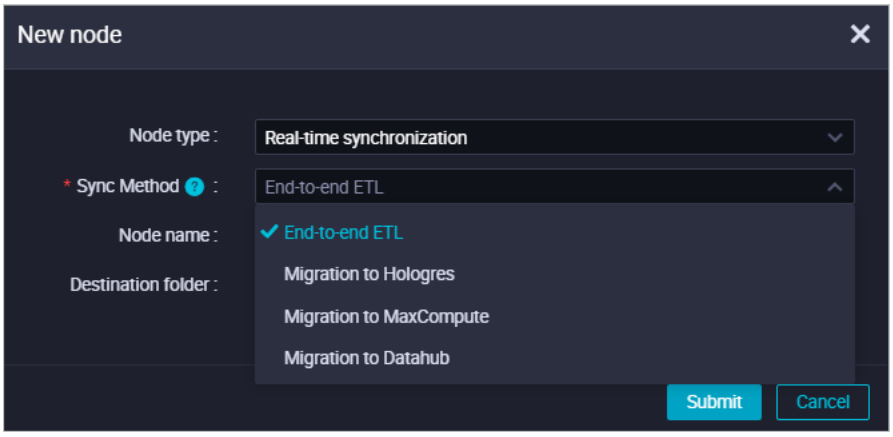
The real-time synchronization feature is in public preview. This feature is available in the following regions: China (Hangzhou), China (Shanghai), China (Beijing), China (Zhangjiakou), China (Shenzhen), and China (Chengdu).

Create a real-time sync node

1. Log on to the DataWorks console.
2. In the left-side navigation pane, click **Workspaces**.
3. In the top navigation bar, select the region in which the workspace that you want to manage resides. Find the workspace and click **Data Development** in the Actions column.
4. On the Data Development tab, move the pointer over the +Create icon and choose **Data integration > Real-time synchronization**.

   Alternatively, you can click a workflow in the Business process section, right-click **Data Integration**, and then choose **New > Real-time synchronization**. For more information about the data stores that the real-time synchronization feature supports, see Plug-ins for data sources that support real-time synchronization.

5. In the **New node** dialog box, set the parameters as required.



New node

Node type : Real-time synchronization

* Sync Method : End-to-end ETL

Node name :

Destination folder :

Parameter	Description
Node type	The type of the node. Default value: **Real-time synchronization**.
Sync Method	The method for synchronizing data. Valid values: ○ **End-to-end ETL**: synchronizes data in one table to one or more tables. Data transformation is supported during the synchronization process. ○ **Migration to Hologres**: synchronizes all or some tables in a database to Hologres. Destination tables can be automatically created in Hologres. ○ **Migration to MaxCompute**: synchronizes all or some tables in a database to MaxCompute. ○ **Migration to DataHub**: synchronizes all or some topics in a database to DataHub.
Node name	The name of the node. The node name must be 1 to 128 characters in length, and can contain letters, digits, underscores (_), and periods (.).
Destination folder	The folder where the node resides.

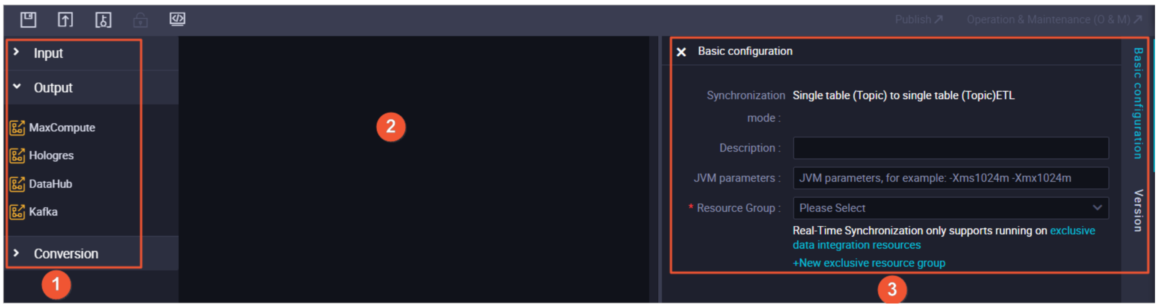
6. Click **Submit**.

Configure the real-time sync node

The operations that you can perform on the configuration tab of the real-time sync node vary based on the synchronization method you selected.

- To configure the real-time sync node for which **Sync Method** is set to **End-to-end ETL**, perform the following steps:

  i. Double-click the real-time sync node. On the node configuration tab that appears, click the **Basic configuration** tab in the right-side navigation pane. On the Basic configuration tab, select the desired resource group from the **Resource Group** drop-down list.



Basic configuration

Synchronization mode : Single table (Topic) to single table (Topic)ETL

Description :

JVM parameters : JVM parameters, for example: -Xms1024m -Xmx1024m

* Resource Group : Please Select

Real-Time Synchronization only supports running on exclusive data integration resources

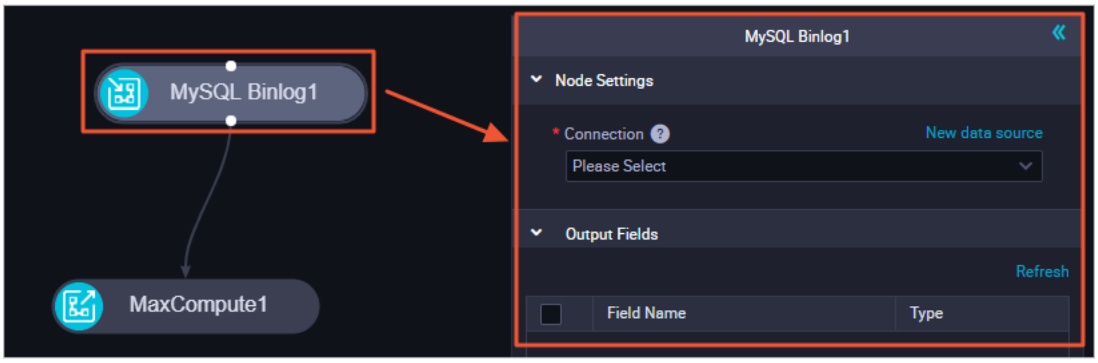
+New exclusive resource group

No.	Description
1	The left-side navigation tree. This pane consists of the **Input**, **Output**, and **Conversion** sections.

No.	Description
2	The configuration canvas of the real-time sync node. You can drag components from the navigation tree to the canvas.
3	The property configuration pane of the real-time sync node. This pane appears after you click a node on the canvas or click the **Basic configuration** tab in the right-side navigation pane. **Notice** You must select a resource group before you commit the node. Otherwise, the system returns an error when you commit the node. Real-time sync nodes can be run only on an exclusive resource group for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.

ii. Drag components from the navigation tree to the canvas, and drag directed lines to connect the nodes on the canvas. Data will be synchronized from upstream nodes to downstream nodes based on the connection.

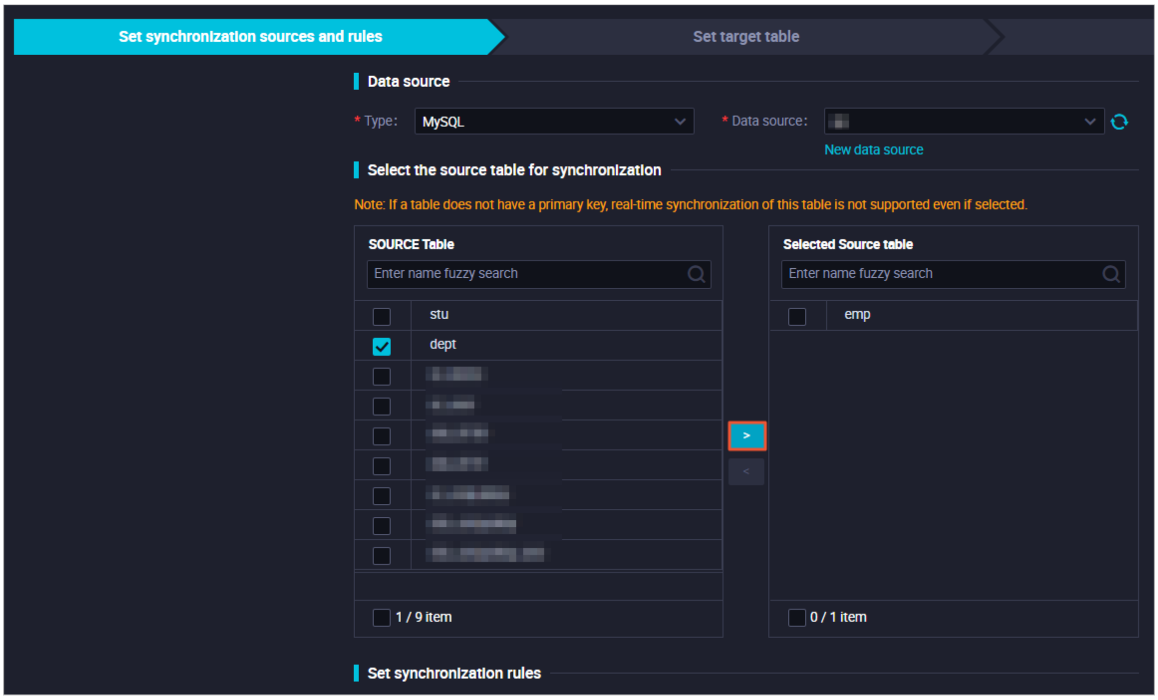
iii. Click each node. In the configuration pane that appears, set the required parameters in the **Node configuration** section. For more information, see Supported data stores.



iv. Click the 💾 icon in the toolbar.

- To configure the real-time sync node for which **Sync Method** is set to **Migration to Hologres**, perform the following steps:

  i. Double-click the real-time sync node. On the node configuration tab that appears, click the **Basic configuration** tab in the right-side navigation pane. On the Basic configuration tab, select the desired resource group from the **Resource Group** drop-down list.



Data source

* Type: MySQL

* Data source: [redacted]

New data source

Select the source table for synchronization

Note: If a table does not have a primary key, real-time synchronization of this table is not supported even if selected.

SOURCE Table

Enter name fuzzy search

□ stu

☑ dept

□ [redacted]

□ [redacted]

□ [redacted]

□ [redacted]

□ [redacted]

□ [redacted]

□ [redacted]

□ 1 / 9 item

Selected Source table

Enter name fuzzy search

□ emp

□ 0 / 1 item

Set synchronization rules

> **Notice** You must select a resource group before you commit the node. Otherwise, the system returns an error when you commit the node. Real-time sync nodes can be run only on an exclusive resource group for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.

ii. In the **Data source** section, set the **Type** and **Data source** parameters.

iii. In the **Select the source table for synchronization** section, select the tables to be synchronized in the **SOURCE Table** list and click the > icon to move the tables to the **Selected Source table** list.

The SOURCE Table list displays all the tables in the source database. You can select all or some tables to synchronize them at a time.

> **Notice** If a selected table does not have a primary key, the table cannot be synchronized in real time.

iv. (Optional)In the **Set synchronization rules** section, click **Add rule** and select an option to configure naming rules for destination tables.

Supported options include **Table name conversion rules** and **Target table name rule**.

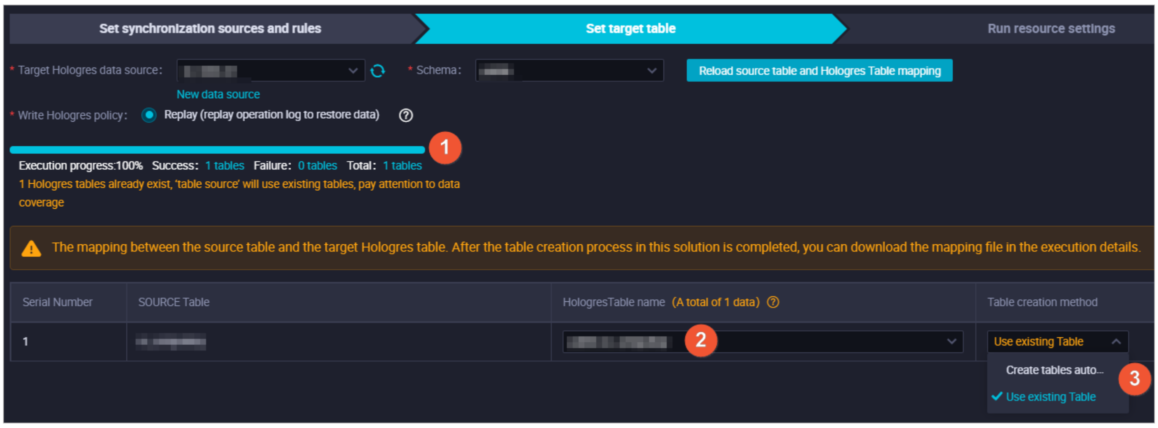
- **Table name conversion rules**: the rules for converting the names of source tables to that of destination tables.
- **Target table name rule**: the rule for adding a prefix and suffix to the converted names of destination tables.

v. Click **Next Step**.

vi. In the **Set target table** step, set the **Target Hologres data source** and **Schema** parameters.

vii. Click **Reload source table and Hologres Table mapping** to configure the mappings between the source tables and destination Hologres tables.

viii. Check the source and destination tables after the mappings are created, and click **Next Step**.



Set synchronization sources and rules | Set target table | Run resource settings

* Target Hologres data source: [redacted]
New data source

* Schema: [redacted]

Reload source table and Hologres Table mapping

* Write Hologres policy: ● Replay (replay operation log to restore data)

Execution progress:100% Success: 1 tables Failure: 0 tables Total: 1 tables

1 Hologres tables already exist, 'table source' will use existing tables, pay attention to data coverage

The mapping between the source table and the target Hologres table. After the table creation process in this solution is completed, you can download the mapping file in the execution details.

Serial Number	SOURCE Table	HologresTable name (A total of 1 data)	Table creation method
1	[redacted]	[redacted]	Use existing Table

No.	Description
1	The mapping progress between the source and destination tables. **Note** The mapping may take a long period of time if the number of source tables to be synchronized is large.
2	The destination tables to which data is written. The tables can be existing ones or the ones that are automatically created. **Note** An error message appears if the selected source table does not have a primary key. The synchronization can be performed if one of the selected source tables has a primary key. Source tables without primary keys are ignored during the synchronization.
3	The method of creating a destination table. The message that appears in the **Hologres Table name** column varies depending on the method that you select. ■ If you select **Create tables automatically**, the **Create tables automatically** dialog box appears after you click **Next Step**. Click **Start table building** in the dialog box, and then click **Close** after the table is created. You can click the table name to view and modify the table creation statements. ■ If you select **Use existing Table**, you must select a table from the drop-down list in the **Hologres Table name** column.

ix. In the **Run resource settings** step, set the **Maximum number of connections supported by source read** and **Number of concurrent writes on the target side** parameters and then click the 💾 icon in the toolbar.

- To configure the real-time sync node for which **Sync Method** is set to **Migration to MaxCompute**, perform the following steps:

  i. Double-click the real-time sync node. On the node configuration tab that appears, click the **Basic configuration** tab in the right-side navigation pane. On the Basic configuration tab, select the desired resource group from the **Resource Group** drop-down list.

  ii. In the **Data source** section, set the **Type** and **Data source** parameters.

  iii. In the **Select the source table for synchronization** section, select the tables to be synchronized in the **SOURCE Table** list and click the > icon to move the tables to the **Selected Source table** list.

  The SOURCE Table list displays all the tables in the source database. You can select all or some tables to synchronize them at a time.

  Notice If a selected table does not have a primary key, the table cannot be synchronized in real time.

  iv. In the **Set synchronization rules** section, click **Add rule** and select an option to configure naming rules for destination tables.

Supported options include **Table name conversion rules** and **Target table name rule**.

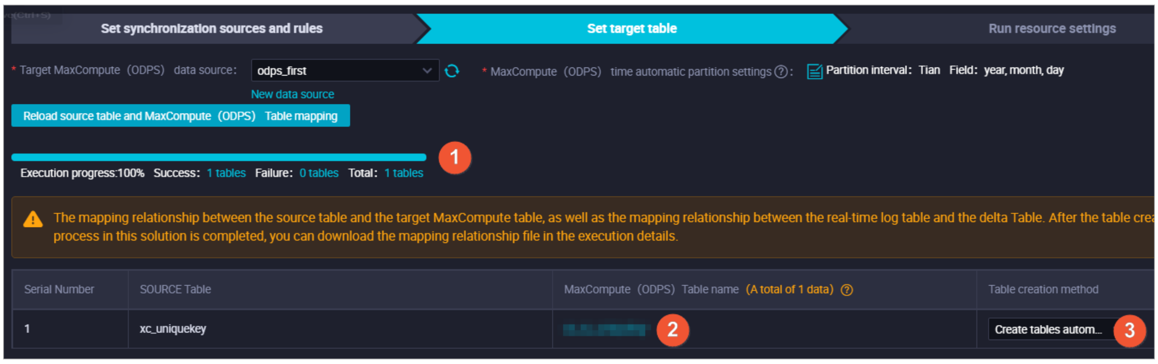
- **Table name conversion rules**: the rules for converting the names of source tables to that of destination tables.
- **Target table name rule**: the rule for adding a prefix and suffix to the converted names of destination tables.

v. Click **Next Step**.

vi. In the **Set target table** step, select a connection from the **Target MaxCompute (ODPS) data source** drop-down list and click the 📝 icon next to **MaxCompute (ODPS) time automatic partition settings**. In the **Edit** dialog box, set the partition interval of tables in MaxCompute to day or hour.

vii. Click **Reload source table and MaxCompute (ODPS) Table mapping** to configure the mappings between the source tables and destination MaxCompute tables.

viii. Check the source and destination tables after the mappings are created, and click **Next Step**.



Set synchronization sources and rules
Set target table
Run resource settings

* Target MaxCompute (ODPS) data source: odps_first

New data source
Reload source table and MaxCompute (ODPS) Table mapping

* MaxCompute (ODPS) time automatic partition settings: Partition interval: Tian Field: year, month, day

Execution progress:100% Success: 1 tables Failure: 0 tables Total: 1 tables

The mapping relationship between the source table and the target MaxCompute table, as well as the mapping relationship between the real-time log table and the delta Table. After the table crea process in this solution is completed, you can download the mapping relationship file in the execution details.

Serial Number	SOURCE Table	MaxCompute (ODPS) Table name (A total of 1 data)	Table creation method
1	xc_uniquekey	[redacted] 2	Create tables autom... 3

No.	Description
1	The mapping progress between the source and destination tables. **Note** The mapping may take a long period of time if the number of source tables to be synchronized is large.
2	The destination tables to which data is written. The tables can be existing ones or the ones that are automatically created. **Note** An error message appears if the selected source table does not have a primary key. The synchronization can be performed if one of the selected source tables has a primary key. Source tables without primary keys are ignored during the synchronization.
3	The method of creating a destination table. The message that appears in the **MaxCompute (ODPS) Table name** column varies depending on the method that you select. ■ If you select **Create tables automatically**, the **Create tables automatically** dialog box appears after you click **Next Step**. Click **Start table building** in the dialog box, and then click **Close** after the table is created. You can click the table name to view and modify the table creation statements. ■ If you select **Use existing Table**, you must select a table from the drop-down list in the **MaxCompute (ODPS) Table name** column.

ix. In the **Run resource settings** step, set the **Maximum number of connections supported by source read** and **Number of concurrent writes on the target side** parameters and then click the 💾 icon in the toolbar.

- To configure the real-time sync node for which **Sync Method** is set to **Migration to DataHub**, perform the following steps:

  i. Double-click the real-time sync node. On the node configuration tab that appears, click the **Basic configuration** tab in the right-side navigation pane. On the Basic configuration tab, select the desired resource group from the **Resource Group** drop-down list.

  ii. In the **Data source** section, set the **Type** and **Data source** parameters.

iii. In the **Select the source table for synchronization** section, select the tables to be synchronized in the **SOURCE Table** list and click the > icon to move the tables to the **Selected Source table** list.

The SOURCE Table list displays all the tables in the source database. You can select all or some tables to synchronize them at a time.

> **Notice** If a selected table does not have a primary key, the table cannot be synchronized in real time.

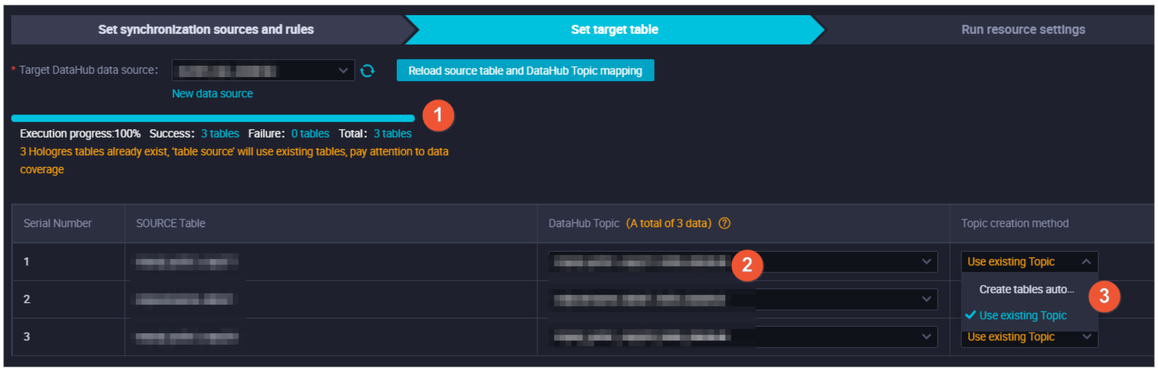
iv. In the **Set synchronization rules** section, click **Add rule** and then select an option to configure naming rules for destination tables.

Supported options include **SOURCE table name and Topic conversion rules** and **Target Topic rules**.

v. Click **Next Step**.

vi. In the **Set target table** step, select a connection from the **Target DataHub data source** drop-down list and then click **Reload source table and DataHub Topic mapping** to configure the mappings between the source tables and destination DataHub topics.

vii. Check the source tables and destination topics after the mappings are created, and click **Next Step**.



* Target DataHub data source: [redacted]

Execution progress:100% Success: 3 tables Failure: 0 tables Total: 3 tables

3 Hologres tables already exist, 'table source' will use existing tables, pay attention to data coverage

Serial Number	SOURCE Table	DataHub Topic (A total of 3 data)	Topic creation method
1	[redacted]	[redacted]	Use existing Topic
2	[redacted]	[redacted]	
3	[redacted]	[redacted]	Use existing Topic

No.	Description
1	The mapping progress between the source tables and destination topics. **Note** The mapping may take a long period of time if the number of source tables to be synchronized is large.
2	The destination topics to which data is written. The topics can be existing ones or the ones that are automatically created.
3	The method of creating a destination topic. The message that appears in the **Topic** column varies depending on the method that you select. ▪ If you select **Create tables automatically**, the **Create tables automatically** dialog box appears after you click **Next Step**. Click **Start table building** in the dialog box, and then click **Close** after the topic is created. ▪ If you select **Use existing Topic**, you must select a topic from the drop-down list in the **Topic** column.

viii. In the **Run resource settings** step, set the **Maximum number of connections supported by source read** and **Number of concurrent writes on the target side** parameters and then click the 💾 icon in the toolbar.

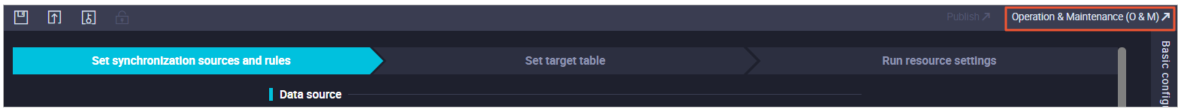
Commit the real-time sync node

1. On the configuration tab of the real-time sync node, click the icon in the toolbar.
2. In the **Submit New version** dialog box, enter your comments in the **Change description** field.
3. Click **OK**.

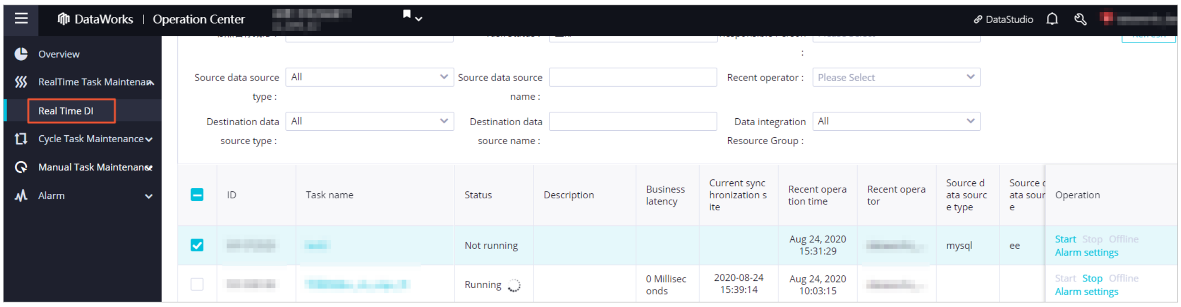
   In a workspace in standard mode, you must click **Publish** in the upper-right corner after you commit the real-time sync node. For more information, see Deploy a node.

Manage the real-time sync node

1. After you commit or deploy the real-time sync node, click **Operation & Maintenance (O & M)** in the upper-right corner of the node configuration tab to manage the node on the **Real Time DI** page.


Publish
Operation & Maintenance (O & M)
Set synchronization sources and rules
Set target table
Run resource settings
Data source
Basic config

2. On the **Real Time DI** page, find the real-time sync node, click the node name, and then view the O&M details about the node.



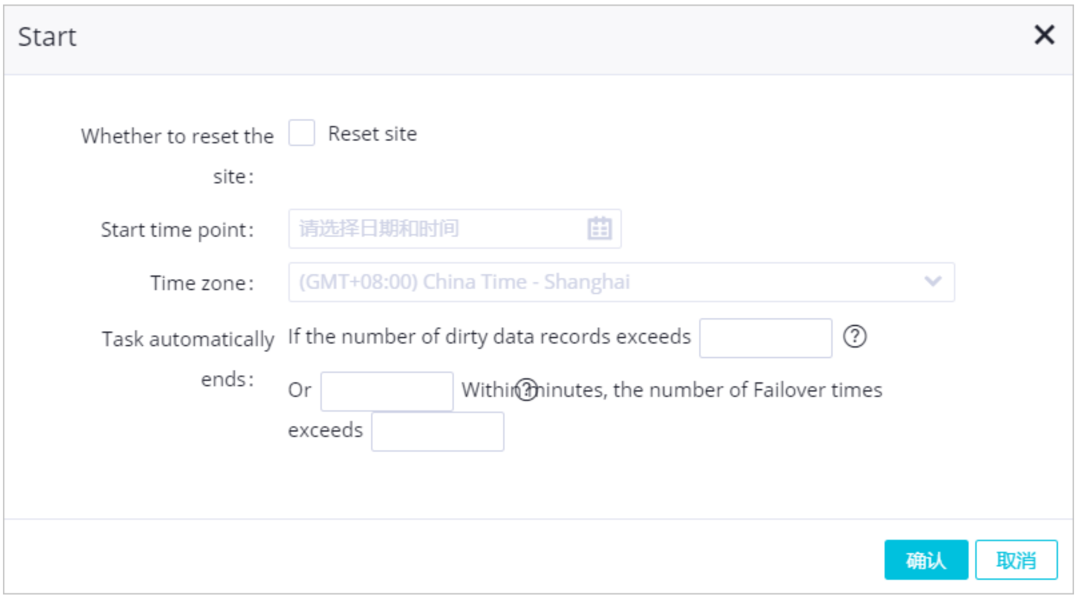
Source data source type : All
Source data source name :
Recent operator : Please Select
Destination data source type : All
Destination data source name :
Data integration Resource Group : All

☐	ID	Task name	Status	Description	Business latency	Current synchronization site	Recent operation time	Recent operator	Source data source type	Source data source	Operation
☑	[redacted]	[redacted]	Not running				Aug 24, 2020 15:31:29	[redacted]	mysql	ee	Start Stop Offline Alarm settings
☐	[redacted]	[redacted]	Running		0 Milliseconds	2020-08-24 15:39:14	Aug 24, 2020 10:03:15	[redacted]			Start Stop Offline Alarm settings

On this page, you can start, stop, undeploy, or configure alert settings for the real-time sync node.

- To start a node that is not running, perform the following steps:
  a. Find the node and click **Start** in the Operation column.

b. In the **Start** dialog box, set the parameters as required.



Start

Whether to reset the site: □ Reset site

Start time point: 请选择日期和时间

Time zone: (GMT+08:00) China Time - Shanghai

Task automatically ends: If the number of dirty data records exceeds

Or Within minutes, the number of Failover times exceeds

Parameter	Description
Whether to reset the site	Specifies whether to set the time point for next startup. If you select Reset site, the **Start time point** and **Time zone** parameters are required.
Start time point	The date and time for starting the real-time sync node.
Time zone	The time zone where the source data store resides. Select a time zone from the **Time zone** drop-down list.
Task automatically ends	■ The condition for automatically terminating the real-time sync node. You can specify the maximum number of dirty data records allowed. If you set the value to 0, no dirty data records are allowed. If the value is empty, the node continues no matter whether dirty data records exist. ■ You can also specify the maximum number of failover times. If the value is empty, the node is automatically terminated if the node fails for 100 times every 5 minutes. This avoids resource occupation caused by frequent startup.

c. Click **OK**.

- To stop a running node, perform the following steps:
  a. Find the node and click **Stop** in the Operation column.
  b. In the message that appears, click **Stop**.
- To bring offline a node that is not running, perform the following steps:
  a. Find the node and click **Offline** in the Operation column.
  b. In the message that appears, click **Offline**.
- Find the node and click **Alarm settings** in the Operation column. On the page that appears, you can view alert event information and alert rules on the **Alert event** and **Alarm rules** tabs.
- To configure alert settings for a node, perform the following steps:
  a. Select the node and click **New Alarm** in the lower part of the page.

   b. In the **New rule** dialog box, set the parameters as required.

Parameter	Description
Name	Required. The name of the rule to create.
Description	The description of the rule.
Indicators	The indicators in the rule to create. Valid values: **Task Status**, **Business latency**, **Failover**, **Dirty Data**, and **DDL error**.
Threshold	The threshold for reporting an alert. The default value is 5 minutes for both **WARNING** and **CRITICAL** alerts.
Alarm interval	The interval for reporting alerts. The default value is 5 minutes.
WARNING	The methods for sending alerts. Valid values: **Mail**, **SMS**, and **DingTalk**. **Note** Only Singapore, Malaysia(Kuala Limpur), and Germany(Frankfurt) support the SMS reminding method. To use the SMS reminding method in other regions, submit a ticket to contact DataWorks technical support.
CRITICAL	
Receiver	The person who receives alerts. Select a receiver from the **Receiver** drop-down list.

   c. Click **OK**.

- To modify alert settings for a node, perform the following steps:

   a. Select the node and click **Operation alarm** in the lower part of the page.

   b. In the **Operation alarm** dialog box, set the **Operation type** and **Alarm indicators** parameters.

   DataWorks automatically modifies all the rules for the selected alert types at a time.

   c. Click **OK**.

4.Sync solutions

4.1. Overview

DataWorks provides solutions for various data synchronization scenarios, such as real-time synchronization, full batch synchronization, and incremental batch synchronization. These solutions help you migrate your business data to the cloud in a more efficient and convenient way.

Background information

In actual business scenarios, data cannot be synchronized by using only one or several simple batch or real-time synchronization nodes. Instead, multiple batch synchronization nodes, real-time synchronization nodes, and data processing nodes are required to synchronize data. In this case, complex configurations are required. To resolve this issue, DataWorks provides scenario-based synchronization solutions and allows you to synchronize data between different data sources with simple configurations. For example, you can easily synchronize data to Elasticsearch, Hologres, or MaxCompute by using relevant solutions provided by DataWorks. This simplifies data synchronization.

For example, a large amount of data is stored in your database system, and you want to synchronize full and incremental data from your database to MaxCompute for analysis. The traditional data synchronization method allows you to perform full synchronization or perform incremental synchronization based on fields such as modify_time in database tables. However, these fields may not exist in database tables in an actual scenario. Therefore, you cannot use the Java Database Connectivity (JDBC) driver to extract data for incremental synchronization. The One-click real-time synchronization to MaxCompute solution allows you to synchronize full and incremental data in your database to MaxCompute in real time. After the synchronization, the full and incremental data is automatically merged in MaxCompute. This simplifies data synchronization.

Synchronization solutions provide the following benefits:

- Initializes full data.
- Writes incremental data in real time.
- Automatically merges full and incremental data at a scheduled time and writes the data to the new partition of a full table.

Limits

Synchronization solutions provided by DataWorks do not support data synchronization across time zones. If the time zone where data sources in a synchronization solution reside is different from the time zone of the resource group that is used to run the solution, errors may occur during data synchronization.

Supported data sources

The following table describes the data sources supported by the synchronization solutions of DataWorks.

Destination	Source	References for configuring data sources	References for configuring synchronization nodes
Elasticsearch	• MySQL • PolarDB for MySQL ? **Note** Among PolarDB data sources, only PolarDB for MySQL data sources are supported.	Configure data sources for data synchronization from MySQL	Configure and view a batch synchronization solution used to synchronize all data in a database

Destination	Source	References for configuring data sources	References for configuring synchronization nodes
Hologres	• PolarDB for MySQL • Oracle • MySQL • PolarDB-X	• Configure a data source (PolarDB) • Configure a source Oracle data source • Configure data sources for data synchronization from MySQL • Configure data sources for data synchronization from PolarDB-X	Create and configure a sync solution
MaxCompute	• PolarDB for MySQL • Oracle • MySQL • PolarDB-X	• Configure a source PolarDB data source • Configure data sources for data synchronization from Oracle • Configure data sources for data synchronization from MySQL	Synchronize data to MaxCompute in real time

Resource usage and billing

When you synchronize data, Data Integration nodes run on resources in resource groups for Data Integration and resource groups for scheduling. You can use only exclusive resource groups for Data Integration. Before you synchronize data, you must purchase an exclusive resource group for Data Integration and add the exclusive resource group to your DataWorks workspace.

The following table describes the performance metrics of exclusive resource groups for Data Integration.

Specifications	Maximum number of parallel threads for a batch synchronization node	Maximum number of parallel real-time synchronization nodes for a single table in a source	Maximum number of parallel real-time synchronization nodes for multiple tables in a source	Maximum number of parallel real-time synchronization nodes for table shards
4c8g	8	3	3	Not supported
8c16g	16	6	6	1
12c24g	24	9	9	1
16c32g	32	12	12	2
24c48g	48	18	18	3

For information about the pricing of exclusive resource groups for Data Integration in different regions, see Pricing. The actual prices on the buy page prevail.

You can estimate the required resources and purchase an exclusive resource group for Data Integration based on the amount of data that you want to synchronize. For more information about exclusive resource groups for Data Integration, see Exclusive resources for Data Integration. You can use exclusive resource groups for scheduling or the shared resource group for scheduling to run nodes.

Note

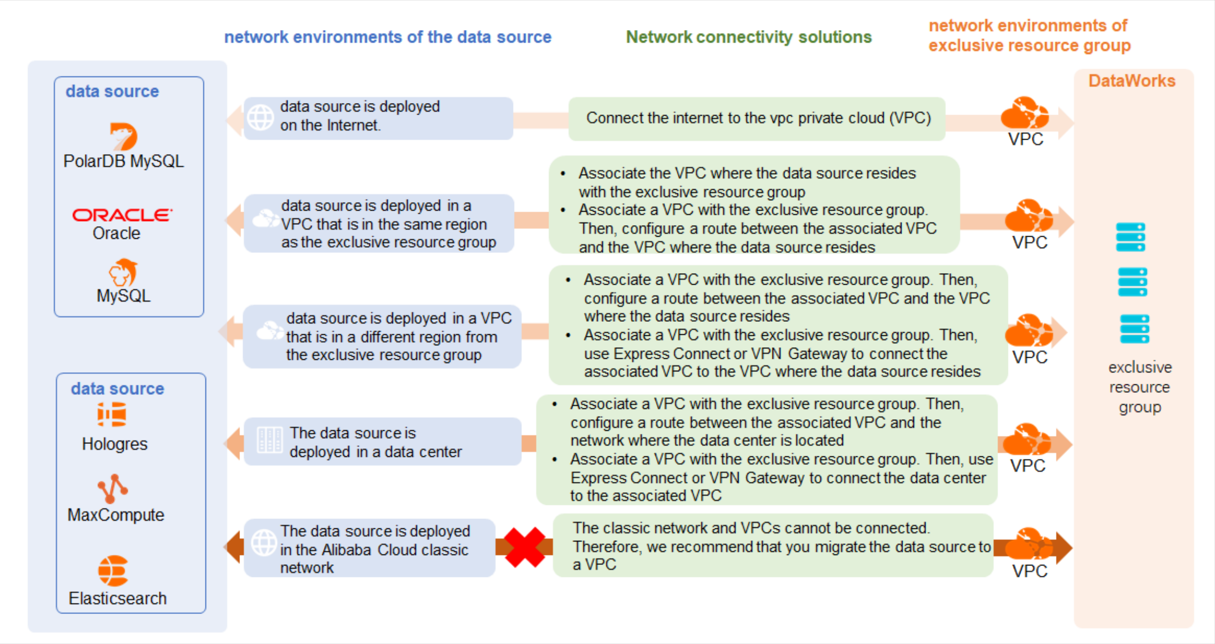
- You are not charged for synchronization solutions. However, a synchronization solution consists of multiple nodes, and you are charged for the resources used to run the nodes. For example, exclusive resource groups for Data Integration and resource groups for scheduling are used to run the real-time and batch synchronization nodes in a synchronization solution. In this case, you are charged for the resource groups.
- Specific nodes in a synchronization solution may consume MaxCompute computing resources. For example, the One-click real-time synchronization to MaxCompute solution requires periodic merging of full and incremental data. The fees for the MaxCompute computing resources are included in your MaxCompute bill, and are positively correlated to the size of the full data and the merging cycle. For more information, see Billing method in MaxCompute documentation.

Network connectivity solutions

For more information about network connectivity solutions, see Overview of network connectivity solutions. This section describes the solutions that can be used to connect a data source to an exclusive resource group.

An exclusive resource group for Data Integration is essentially a group of ECS instances. After you purchase such an exclusive resource group, it is isolated from other services. You must associate the resource group with a virtual private cloud (VPC) to ensure network connectivity between the resource group and data sources during subsequent data synchronization.

The network connectivity solutions vary based on the network environments of the source and destination.



data source is deployed on the Internet.

Connect the internet to the vpc private cloud (VPC)

data source is deployed in a VPC that is in the same region as the exclusive resource group

- Associate the VPC where the data source resides with the exclusive resource group
- Associate a VPC with the exclusive resource group. Then, configure a route between the associated VPC and the VPC where the data source resides

data source is deployed in a VPC that is in a different region from the exclusive resource group

- Associate a VPC with the exclusive resource group. Then, configure a route between the associated VPC and the VPC where the data source resides
- Associate a VPC with the exclusive resource group. Then, use Express Connect or VPN Gateway to connect the associated VPC to the VPC where the data source resides

The data source is deployed in a data center

- Associate a VPC with the exclusive resource group. Then, configure a route between the associated VPC and the network where the data center is located
- Associate a VPC with the exclusive resource group. Then, use Express Connect or VPN Gateway to connect the data center to the associated VPC

The data source is deployed in the Alibaba Cloud classic network

The classic network and VPCs cannot be connected. Therefore, we recommend that you migrate the data source to a VPC

- The data source is deployed on the Internet.

  Connect the data source to the virtual private cloud (VPC) that is associated with the exclusive resource group.

- The data source is deployed in a VPC that is in the same region as the exclusive resource group.
  - Same zone: Associate the exclusive resource group with the VPC in which the data source resides.
  - Different zones: Associate the exclusive resource group with a VPC. Then, configure a route between the associated VPC and the VPC in which the data source resides.
- The data source is deployed in a VPC that is in a different region from the region in which the exclusive resource group resides.
  - Associate the exclusive resource group with a VPC. Then, configure a route between the associated VPC and the VPC in which the data source resides.
  - Associate the exclusive resource group with a VPC. Then, use Express Connect or VPN Gateway to connect the associated VPC to the VPC in which the data source resides.
- The data source is deployed in a data center.

  - Associate the exclusive resource group with a VPC. Then, configure a route between the associated VPC and the network to which the data center is connected.
  - Associate the exclusive resource group with a VPC. Then, use Express Connect or VPN Gateway to connect the network to which the data center is connected to the associated VPC.

- The data source is deployed on the Alibaba Cloud classic network.

  The classic network and VPCs cannot be connected. Therefore, we recommend that you migrate the data source to a VPC.

Procedure

To use a synchronization solution of DataWorks, perform the following steps:

1. Plan and configure resources.

   Estimate the required resources and purchase an exclusive resource group for Data Integration and an exclusive resource group for scheduling based on your network conditions and the amount of data that you want to synchronize. Then, configure resources to ensure network connectivity.

2. Configure data sources.

   After you establish network connections for data sources between which you want to synchronize data, configure the data sources to ensure accessibility. For example, make sure that the IP addresses of the exclusive resource groups are added to the IP address whitelists of the data sources. Otherwise, the synchronization fails.

3. Add data sources.

   Add the data sources to DataWorks as the source and destination. This way, you can associate the data sources when you create a synchronization solution.

4. Create and configure a synchronization solution.

   Create a synchronization solution and set the parameters based on the synchronization scenario.

> Note
>
> - You can add or remove source tables to or from a created synchronization solution. If a real-time synchronization node is running, you must terminate the node before you add or remove the source tables. After you add or remove the tables, click **Submit and Run** to run the solution. DataWorks automatically creates batch synchronization nodes and updates real-time synchronization nodes. For more information about how to add or remove source tables to or from a synchronization solution that is running, see Add or remove source tables to or from a synchronization solution that is running.
> - When you configure a destination table for a synchronization solution, if you select **Create Table** for the **Table creation method** parameter, you can click the name of the table to modify the table creation statements or configurations of the table as needed. Check whether the table creation statements or configurations meet your requirements.

For more information about the synchronization between data sources, see the following topics:

- Synchronize data to Elasticsearch
- Synchronize data to Hologres
- Synchronize data to MaxCompute

4.2. Select a synchronization solution

The Data Integration service of DataWorks allows you to use a synchronization solution to synchronize data from a source to a destination in real time or in batch mode. You can use a synchronization solution to synchronize multiple tables at a time or synchronize both full and incremental data. If you want to synchronize both full and incremental data, you can synchronize the incremental data after the full data is synchronized.

Go to the Tasks page

1. Log on to the DataWorks console.
2. In the left-side navigation pane, click **Workspaces**.
3. In the top navigation bar, select the region in which your workspace resides. Find the workspace and click **Data Integration** in the Actions column.

4. In the left-side navigation pane, click **Synchronization task**. The **Tasks** page appears.

   You can create synchronization nodes and view the status of the created synchronization nodes on this page.

   When you create a synchronization node, you can select a synchronization solution as needed. For more information, see the Select a synchronization solution section of this topic. A synchronization node can be in the following states:

   - **Not Running**: The synchronization node is not run. You can click **Submit and Run** in the Operation column of the synchronization node to run the synchronization node.

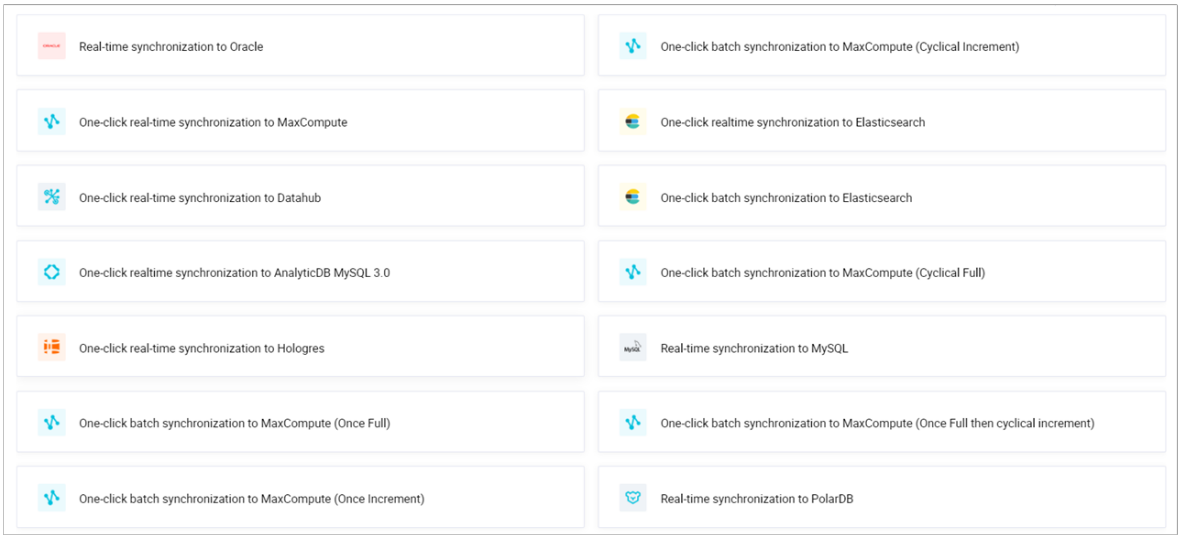
     > ⓘ **Note** You can click **Modify Configuration** to modify a synchronization node only if the synchronization node is in the **Not Running** state. If you click Modify Configuration in the Operation column of a synchronization node in another state, you can view only the information about that synchronization node.

   - **Running**: The synchronization node is running and cannot be terminated. You must wait until the synchronization node is complete.
   - **Exception**: An error occurred while running the synchronization node. You can click **Execution details** in the Operation column of the synchronization node to troubleshoot the error.
   - **Succeeded**: The synchronization node is complete. You can click **Execution details** in the Operation column of the synchronization node to view the results of the synchronization node.

Select a synchronization solution

1. On the Tasks page, click **New task** in the upper-right corner.
2. On the Create Data Synchronization Solution page, select the source and destination for data synchronization, and then select a synchronization solution.

   For more information about the supported source and destination data sources, see Supported data sources. The following figure shows the supported synchronization solutions.



Real-time synchronization to Oracle

One-click batch synchronization to MaxCompute (Cyclical Increment)

One-click real-time synchronization to MaxCompute

One-click realtime synchronization to Elasticsearch

One-click real-time synchronization to Datahub

One-click batch synchronization to Elasticsearch

One-click realtime synchronization to AnalyticDB MySQL 3.0

One-click batch synchronization to MaxCompute (Cyclical Full)

One-click real-time synchronization to Hologres

Real-time synchronization to MySQL

One-click batch synchronization to MaxCompute (Once Full)

One-click batch synchronization to MaxCompute (Once Full then cyclical increment)

One-click batch synchronization to MaxCompute (Once Increment)

Real-time synchronization to PolarDB

   DataWorks supports the following synchronization solutions, which are classified based on the destination data source:

   - **One-click real-time synchronization to DataHub**: used to synchronize data to DataHub.
   - **One-click realtime synchronization to Elasticsearch**: used to synchronize data to Elasticsearch.
   - **One-click real-time synchronization to Hologres**: used to synchronize data to Hologres.
   - **One-click realtime synchronization to AnalyticDB MySQL 3.0**: used to synchronize data to AnalyticDB for MySQL V3.0.
   - **One-click real-time synchronization to Kafka**: used to synchronize data to Kafka.
   - Synchronization solutions for synchronizing data to MaxCompute, including:
     - **One-click real-time synchronization to MaxCompute**
     - **One-click batch synchronization to MaxCompute (Cyclical Full)**
     - **One-click batch synchronization to MaxCompute (Cyclical Increment)**
     - **One-click batch synchronization to MaxCompute (Once Full)**

  - One-click batch synchronization to MaxCompute (Once Increment)
  - One-click batch synchronization to MaxCompute (Once Full then cyclical increment)

4.3. Synchronize data to DataHub

4.3.1. Plan and configure resources

If you use sync solutions of DataWorks to synchronize data, you can use only exclusive resource groups for Data Integration to run Data Integration nodes. However, you can select a shared or exclusive resource group for scheduling based on your business requirements. This topic describes the resources that are used for data synchronization and how to configure the resources.

Context

- Resource planning and preparation

  When you synchronize data, Data Integration nodes run on resources in resource groups for Data Integration and resource groups for scheduling. You can use only exclusive resource groups for Data Integration. Before you synchronize data, you must purchase an exclusive resource group for Data Integration and add the exclusive resource group to your DataWorks workspace.

  For more information about exclusive resource groups for Data Integration, see Exclusive resources for Data Integration.

- Network connectivity

  An exclusive resource group for Data Integration is essentially a group of Elastic Compute Service (ECS) instances. After you purchase such an exclusive resource group, it is isolated from other services. You must associate the resource group with a virtual private cloud (VPC) to ensure network connectivity between the resource group and data sources during subsequent data synchronization.

What's next

After you plan and configure resources, you can configure data sources. You must connect the exclusive resource group for Data Integration to the source and destination. You must also create an account and grant the required permissions to the account. This account is used to access the source and destination. The preceding operations help create a data sync node. For more information about how to configure data sources, see Configure data sources for data synchronization from MySQL, Configure data sources for data synchronization from PolarDB, and Configure data sources for data synchronization from Oracle.

4.3.2. Configure data sources for data synchronization from MySQL

Before you synchronize data from MySQL to DataHub, you can refer to the operations described in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

- Prepare data sources: A MySQL database and a DataHub project are prepared.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.
- Execute the following statement to check the version of your MySQL database. You can use a `MySQL 5.X` or `MySQL 8.X` database.

```
select version();
```

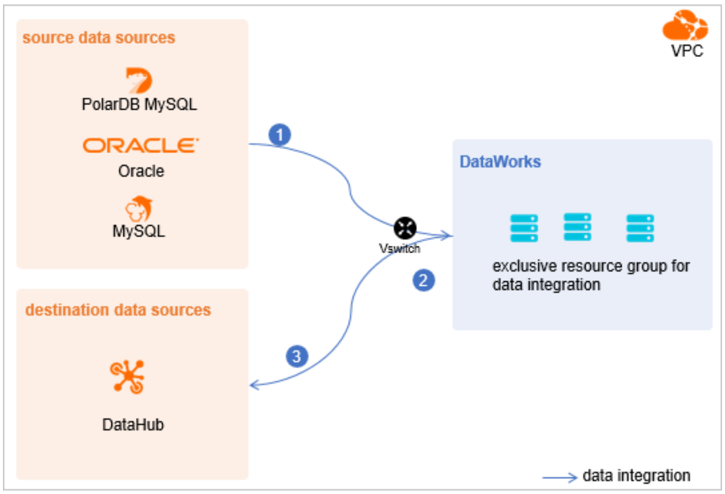
Note Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `v5.x` or `v8.x`. PolarDB for MySQL is not supported. If the MySQL version of your ApsaraDB RDS for MySQL database is not `v5.x` or `v8.x`, use an ApsaraDB RDS for MySQL database whose MySQL version is `v5.x` or `v8.x`. Otherwise, the data synchronization node fails to run.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for data sources

  If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the vSwitch that is bound to the exclusive resource group for Data Integration during network configuration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source data source is a MySQL database, you must enable the binary logging feature. Binary logs record changes to all table schemas and modifications on table data. You can execute statements such as CREATE and ALTER to perform operations on table schemas. You can execute statements such as INSERT, UPDATE, and DELETE to perform operations on table data. You can use binary logs to view the change history of the database, back up incremental data and restore data in the database, and replicate data from the primary database to secondary databases.

  Formats of binary logs:

  - Statement: SQL statement-based replication. Binary logs in this format record the SQL statements that are executed to modify data entries.
  - Row: row-based replication. Binary logs in this format record only modification details about data entries in rows.
  - Mixed: replication in mixed mode. This mode combines the statement and row formats. In most cases, binary logs in the statement format are used to record the SQL statements that are executed to modify data entries, such as functions. If data replication from the primary database to secondary databases cannot be implemented by using binary logs in this format, switch to the row format. MySQL determines which format to use based on each SQL statement that is executed.

Limits

Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.X` or `V8.X` . PolarDB for MySQL is not supported.

Procedure

1. Configure a whitelist for the MySQL database.

   Add the CIDR block of the VPC in which the exclusive resource group resides to the whitelist of the MySQL database.

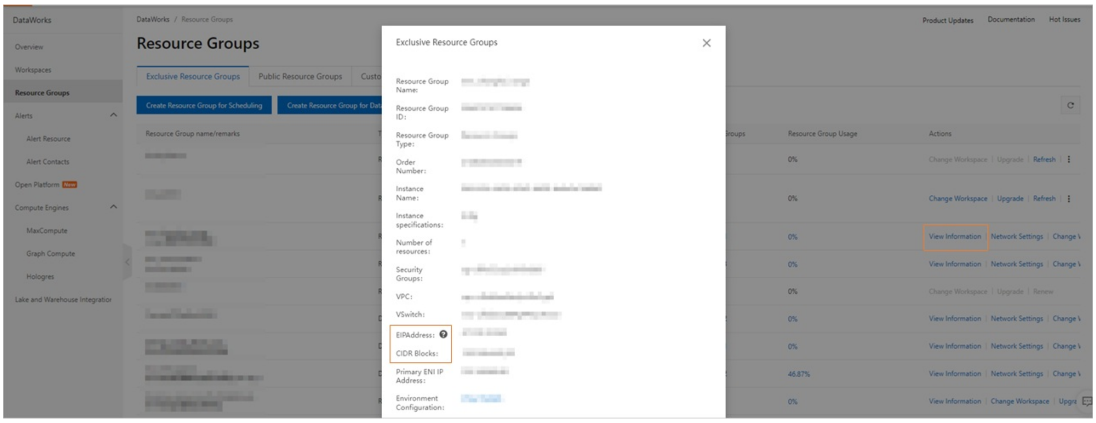
   i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

      a. Log on to the DataWorks console.

      b. In the left-side navigation pane, click **Resource Groups**.

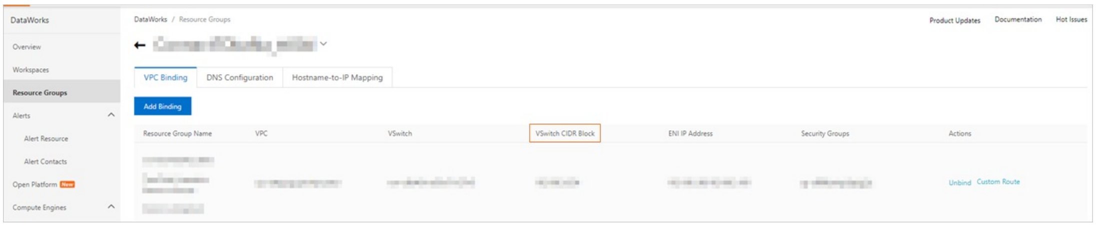
      c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

      d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



      e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

      f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



   ii. Add the EIP and CIDR block of the exclusive resource group to the whitelist of the MySQL database.

2. Create an account and grant permissions to the account.

   You must create an account to log on to the MySQL database. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

   i. Create an account.

      For more information, see Create an account to access a MySQL database.

ii. Grant permissions to the account.

You can run the following command to grant permissions to the account. Alternatively, you can grant the `SUPER` permission to the account. Replace `Account for data synchronization` with the created account when you execute the specific statement.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Password'; // Create an account th
at is used to synchronize data and set a password so that you can use the account and password to acces
s the database from a host. % indicates a host.
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%'; /
/ Grant the SELECT, REPLICATION SLAVE, and REPLICATION CLIENT permissions to the account.
```

`*.*` indicates that the synchronization account is granted the preceding permissions on all tables in all databases. You can also grant the preceding permissions on specified tables in the specified database to the synchronization account. For example, to grant the account the preceding permissions on the *user* table in the *test* database, execute the following statement: `GRANT SELECT, REPLICATION CLIENT ON test.user TO 'Account for data synchronization'@'%';`.

> **Note** The `REPLICATION SLAVE` permission is a global permission. You cannot grant this permission on specified tables in the specified database to the synchronization account.

3. Enable the binary logging feature for the MySQL database.

   Perform the following steps to check whether the binary logging feature is enabled and to query the format of binary logs:

   - Execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_bin";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled.

   - If you use a secondary database to synchronize data, execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_slave_updates";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled for the secondary database.

   If the returned result is different from the preceding result, follow the instructions described in the *MySQL documentation* to enable the binary logging feature.

   Execute the following statement to view the format of binary logs:

   ```
   show variables like "binlog_format";
   ```

   Returned result:

   - *ROW*: The format of binary logs is row.
   - *STATEMENT*: The format of binary logs is statement.
   - *MIXED*: The format of binary logs is mixed.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add data sources.

4.3.3. Configure data sources for data synchronization from PolarDB

Before you synchronize data from PolarDB to DataHub, you can refer to the operations described in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

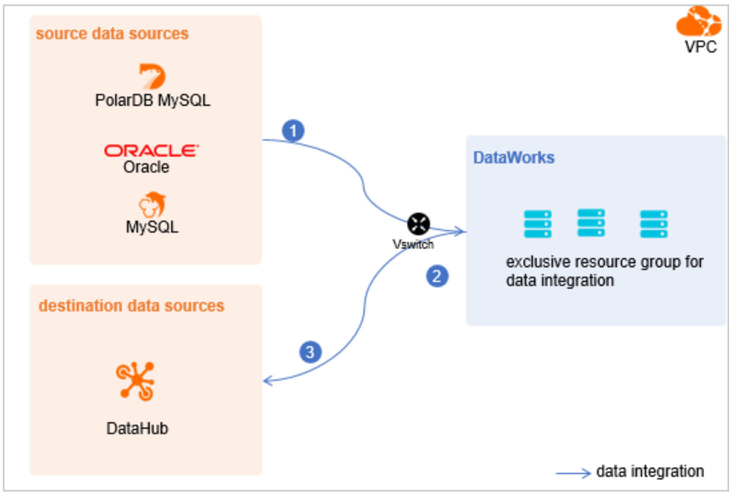
- Prepare data sources: A PolarDB for MySQL cluster and a DataHub project are prepared. In this topic, a PolarDB for MySQL cluster is used as the source.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
    - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
    - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for data sources

  If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can access the data sources.

- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source is a PolarDB for MySQL cluster, you must enable the binary logging feature for the cluster. PolarDB for MySQL is fully compatible with MySQL and uses high-level physical logs to replace binary logs. To facilitate the integration between PolarDB and the MySQL ecosystem, you can enable the binary logging feature for PolarDB clusters.

Limits

- Only PolarDB for MySQL clusters can be used as sources in data synchronization solutions. Other types of PolarDB data sources are not supported. In this topic, PolarDB indicates PolarDB for MySQL data sources.
- Only data stored on the primary node of a PolarDB for MySQL cluster can be synchronized.

Procedure

1. Configure a whitelist for the PolarDB for MySQL cluster.

To add the CIDR block of the VPC where the exclusive resource group for Data Integration resides to a whitelist of the PolarDB for MySQL cluster, perform the following steps:

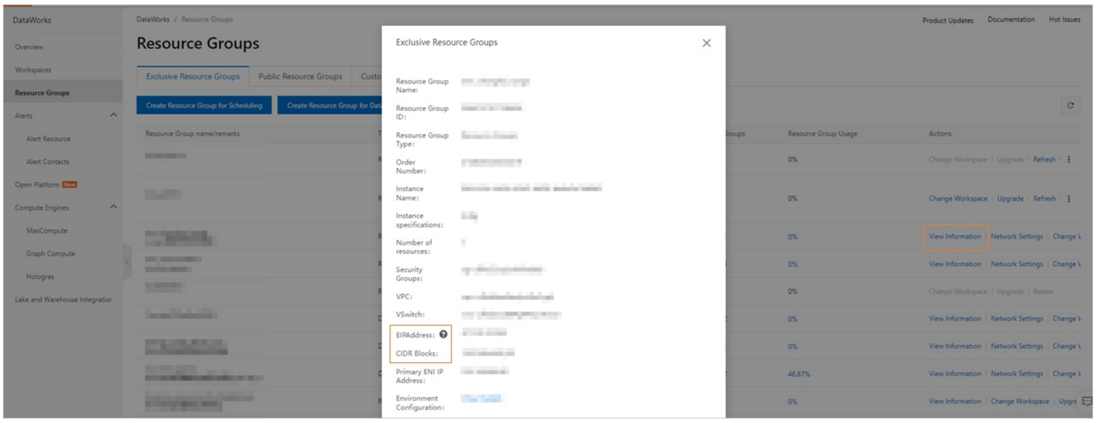
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

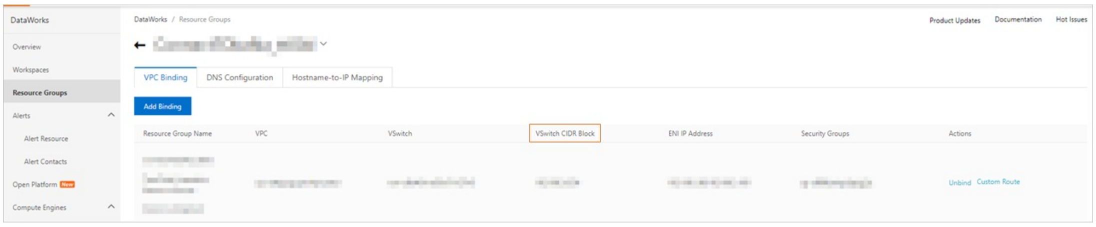
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.

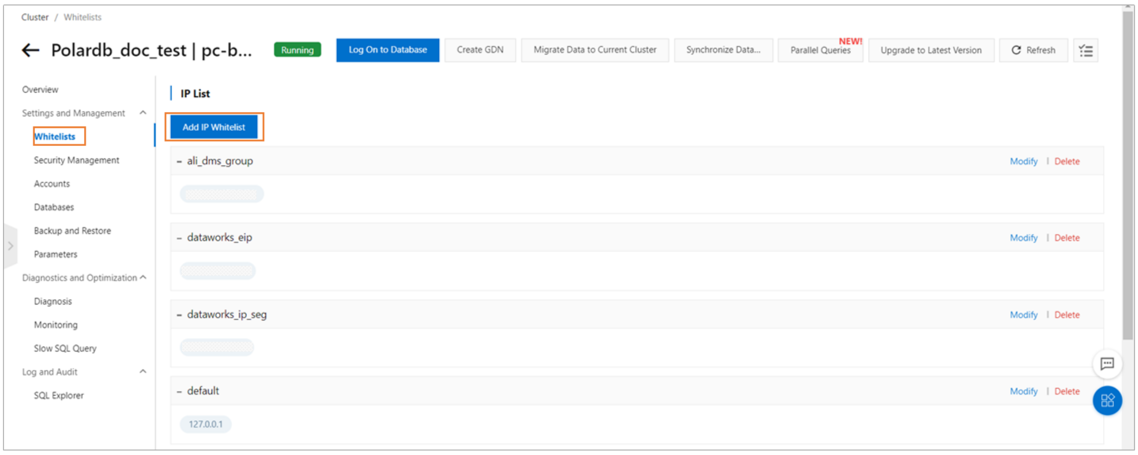


e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR blocks recorded in the preceding steps to the whitelist of the PolarDB for MySQL cluster.



For more information, see Configure an IP whitelist.

2. Create an account and grant the required permissions to the account.

You must create an account to log on to the database of the PolarDB for MySQL cluster. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

i. Create an account.

   For more information, see Create a database account.

ii. Grant the required permissions to the account.

   You can run the following command to grant the required permissions to the account, or you can directly assign the `SUPER` role to the account.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Account for data synchronization';
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%';
```

3. Enable the binary logging feature for the PolarDB for MySQL cluster.

   For more information, see Enable binary logging.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add data sources.

4.3.4. Configure data sources for data synchronization from Oracle

Before you synchronize data from an Oracle data source to DataHub, you can refer to the operations in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions.

Prerequisites

Before you configure the data sources, make sure that the following operations are performed:

- Prepare data sources: An Oracle database and a DataHub project are prepared.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources. The Oracle database version, character encoding formats, and data types must be supported by Data Integration.

-
- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

-
- Enable the generation of database-level archived log files and redo log files and enable database-level supplemental logging

  You must enable database-level redo log files and supplemental logging for the Oracle database to be configured as a source data source.

  -

  - Redo log files: Oracle uses redo log files to ensure that database transactions can be re-executed. This way, data can be recovered in the case of a failure such as a power outage.
  - Supplemental logging: Supplemental logging is used to supplement the information recorded in redo log files. In Oracle, a redo log file is used to record the values of the fields that are modified. Supplemental logging is used to supplement the change history in the redo log file. This ensures that the redo log file contains complete information that describes data changes. If operations such as data recovery and data synchronization are performed, you can view complete statements and data updates. Specific features of the Oracle database can be better implemented after supplemental logging is enabled. Therefore, you must enable supplemental logging for the database.

    For example, if you do not enable supplemental logging, after you execute the UPDATE statement, the redo log file records only the values of the fields that are modified when the UPDATE statement is executed. If you enable supplemental logging, the redo log file records the values of fields before and after a modification. The conditions that are used to modify destination fields are also recorded. When an exception such as a power outage occurs in the database, you can recover data based on the modification details.

    We recommend that you enable supplemental logging for primary key columns or unique index columns.

    - After you enable supplemental logging for primary key columns, the columns that compose a primary key are recorded in logs if the database is updated.
    - After you enable supplemental logging for unique index columns, the columns that compose a unique key or bitmap index are recorded in logs if a column is modified.

- Check character encoding formats

  You must make sure that the Oracle database contains only the character encoding formats that are supported by Data Integration to prevent data synchronization failures. The following character encoding formats are supported by Data Integration: UTF-8, AL32UTF8, AL16UTF16, and ZHS16GBK.

- Check data types

  You must make sure that the Oracle database contains only the data types that are supported by Data Integration to prevent data synchronization failures. The following data types are not supported by Data Integration for real-time data synchronization: LONG, BFILE, LONG RAW, and NCLOB.

Limits

- You can configure the supplemental logging feature only in a primary Oracle database. Supplemental logging can be enabled for a primary or secondary database.
- The following character encoding formats are supported by Data Integration: UTF-8, AL32UTF8, AL16UTF16, and ZHS16GBK.
- The following data types are not supported by Data Integration for real-time data synchronization: LONG, BFILE, LONG RAW, and NCLOB.
-

Procedure

1. Configure a whitelist for the Oracle database.

   Add the CIDR block of the VPC in which the exclusive resource group resides to the whitelist of the Oracle database.

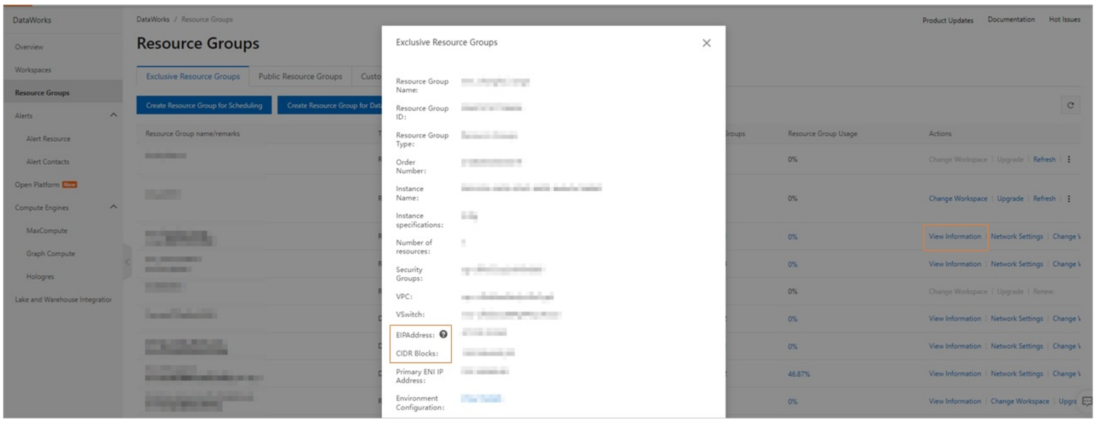
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

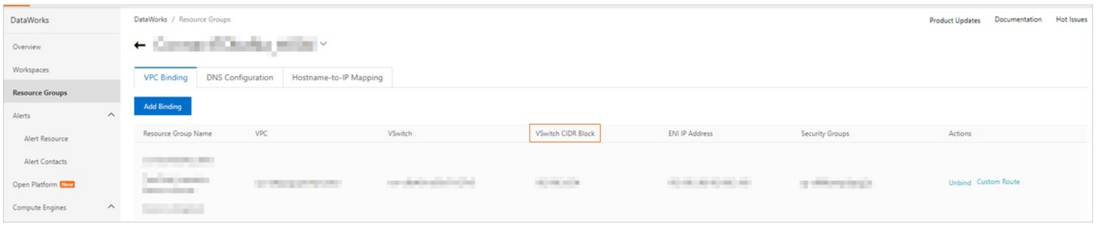
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR block of the exclusive resource group to the whitelist of the Oracle database.

2. Create an account and grant the required permissions to the account.

You must create an account to log on to the database. The account must have the required permissions on the Oracle database.

i. Create an account.

For more information, see Create an Oracle database account.

ii. Grant permissions to the account.

You can run the following commands to grant permissions to the account. Replace `Account for data synchronization` with the created account when you execute the statements.

```
grant create session to 'Account for data synchronization';  // Authorize the account to access the database.
grant connect to 'Account for data synchronization';  // Authorize the account to connect to the database.
grant select on nls_database_parameters to 'Account for data synchronization';  // Authorize the account to query the settings of nls_database_parameters.
grant select on all_users to 'Account for data synchronization';  // Authorize the account to query all users in the database.
grant select on all_objects to 'Account for data synchronization';  // Authorize the account to query all objects in the database.
grant select on DBA_MVIEWS to 'Account for data synchronization';  // Authorize the account to check the materialized view of the database.
grant select on DBA_MVIEW_LOGS to 'Account for data synchronization';  // Authorize the account to view the materialized view logs of the database.
grant select on DBA_CONSTRAINTS to 'Account for data synchronization';  // Authorize the account to view the constraints on all tables of the database.
grant select on DBA_CONS_COLUMNS to 'Account for data synchronization';  // Authorize the account to view information about all columns under specified constraints on all tables of the database.
grant select on all_tab_cols to 'Account for data synchronization';  // Authorize the account to view information about columns in tables, views, and clusters of the database.
grant select on sys.obj$ to 'Account for data synchronization';  // Authorize the account to view objects in the database. sys.obj$ indicates an object table that is contained in the data dictionary table. The object table contains all objects.
grant select on SYS.COL$ to 'Account for data synchronization';  // Authorize the account to view definitions of columns in tables of the database. SYS.COL$ stores column definitions.
grant select on sys.USER$ to 'Account for data synchronization';  // Authorize the account to view the system table of the database. sys.USER$ indicates a default user session service.
grant select on sys.cdef$ to 'Account for data synchronization';  // Authorize the account to view the system table of the database.
grant select on sys.con$ to 'Account for data synchronization';  // Authorize the account to view the constraints of the database. sys.con$ records the constraints.
grant select on all_indexes to 'Account for data synchronization';  // Authorize the account to view all indexes of the database.
grant select on v_$database to 'Account for data synchronization';  // Authorize the account to check the v_$database view of the database.
grant select on V_$ARCHIVE_DEST to 'Account for data synchronization';  // Authorize the account to check the V_$ARCHIVE_DEST view of the database.
grant select on v_$log to 'Account for data synchronization';  // Authorize the account to check the v_$log view of the database. v_$log displays log information about control files.
grant select on v_$logfile to 'Account for data synchronization';  // Authorize the account to check the v_$logfile view of the database. v_$logfile contains information about redo log files.
grant select on v_$archived_log to 'Account for data synchronization';  // Authorize the account to check the v$archived_log view of the database. v$archived_log contains information about archived logs.
grant select on V_$LOGMNR_CONTENTS to 'Account for data synchronization';  // Authorize the account to check the V_$LOGMNR_CONTENTS view of the database.
grant select on DUAL to 'Account for data synchronization';   // Authorize the account to view the DUAL table of the database. DUAL is a virtual table that contains SELECT syntax rules. In Oracle, only one record is retained in the DUAL table.
grant select on v_$parameter to 'Account for data synchronization';  // Authorize the account to check the v_$parameter view of the database. v$parameter is a dynamic dictionary table that stores the values of parameters in the database.
grant select any transaction to 'Account for data synchronization';  // Authorize the account to view transactions of the database.
grant execute on SYS.DBMS_LOGMNR to 'Account for data synchronization';  // Authorize the account to use the LOGMNR tool. The LOGMNR tool helps you analyze transactions and retrieve lost data.
grant alter session to 'Account for data synchronization';  // Authorize the account to modify connection configurations of the database.
grant select on dba_objects to 'Account for data synchronization';  // Authorize the account to view all objects of the database.
grant select on v_$standby_log to 'Account for data synchronization';  // Authorize the account to check the v_$standby_log view of the database. v_$standby_log contains archived logs of the secondary database.
grant select on v_$ARCHIVE_GAP to 'Account for data synchronization';  // Authorize the account to query missing archived logs.
```

To synchronize full data in batch mode, you must also run the following command to grant the query permission on all tables to the account:

```
grant select any table to 'Account for data synchronization';
```

In Oracle 12c or a later version, you must run the following command to grant the log mining permission to the account. The log mining feature is built in Oracle versions earlier than 12c. You do not need to run the command in these versions.

```
grant LOGMINING TO 'Account for data synchronization';
```

3.

4. Check character encoding formats of the database.

   Run the following command to check character encoding formats of the database:

   ```
   select * from v$nls_parameters where PARAMETER IN ('NLS_CHARACTERSET', 'NLS_NCHAR_CHARACTERSET');
   ```

   - v$nls_parameters stores values of parameters in the database.
   - NLS_CHARACTERSET indicates a database character set. NLS_NCHAR_CHARACTERSET indicates a national character set. These two sets are used to store data of the character type.

   The following character encoding formats are supported by Data Integration: UTF-8, AL32UTF8, AL16UTF16, and ZHS16GBK. If the database contains character encoding formats that are not supported by Data Integration, change the formats before you synchronize data.

5. Check the data types of tables in the database.

   You can execute the SELECT statement to query the data types of tables in the database. Sample statement that is executed to query the data types of the *'tablename'* table:

   ```
   select COLUMN_NAME,DATA_TYPE from all_tab_columns where TABLE_NAME='tablename';
   ```

   - COLUMN_NAME: the name of the column.
   - DATA_TYPE: the data type of the column.
   - all_tab_columns: the view that stores information about all columns in tables of the database.
   - TABLE_NAME: the name of the table to query. When you execute the preceding statement, replace *'tablename'* with the name of the table to query.

   You can also execute the `select * from 'tablename';` statement to query the information about the table and obtain the data types of columns.

   The following data types are not supported by Data Integration for real-time data synchronization: LONG, BFILE, LONG RAW, and NCLOB. If a table contains one of these data types, remove the table from the real-time sync solution or change the data type before you synchronize data.

What's next

After the data sources are configured, the source data source, destination data source, and exclusive resource group for Data Integration are connected, and you can use the authorized account to access the data sources. You can add the source data source and destination data source to DataWorks, and associate them with a sync solution when you create the solution.

For more information about how to add a data source, see Add data sources.

4.3.5. Add data sources

Before you configure a real-time data sync node to synchronize data from a data source to the DataHub data source, you must add both data sources to DataWorks for subsequent source and destination configurations.

Prerequisites

Before you add data sources, make sure that the following operations are performed:

- Prepare data sources: A source data source and a destination data source are created.
- Create and grant permission to an account: An account that is used to access data sources is created.

Precautions

DataWorks provides workspaces in basic mode and standard mode. A workspace in basic mode does not isolate the development environment from the production environment. A workspace in standard mode isolates the development environment from the production environment.

If you use a workspace in standard mode, you must separately add data sources to the development environment and production environment.

Add a source MySQL data source

To add a source MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source MySQL data source, see Add a MySQL data source.

Add a source PolarDB for MySQL data source

To add a source PolarDB for MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source PolarDB for MySQL data source, see Add a PolarDB data source.

If the source PolarDB for MySQL data source that you want to add fails the connectivity test, follow the instructions described in What do I do if the PolarDB data source cannot be connected? to handle the exception.

Add a source Oracle data source

To add a source Oracle data source, you must configure information such as the network connection type, the account, and the password as planned. For more information, see Configure data sources for data synchronization from Oracle.

What's next

After you add data sources, you can create and run a data sync node to synchronize data from the source data source to the destination data source.

For more information, see Configure and view a real-time sync solution used to synchronize all data in a database.

4.3.6. Configure and view a real-time sync solution used to synchronize all data in a database

After you configure data sources, network environments, and resource groups, you can create and run a real-time sync solution to synchronize all data in a database. This topic describes how to create a real-time sync solution to synchronize data in some or all tables in a database to DataHub in batch mode and then synchronize incremental data in the database to DataHub in real time. This topic also describes how to view the statuses of the nodes generated by the real-time synchronization solution.

Prerequisites

Before you create a data sync solution, make sure that the following operations are performed:

- Plan and configure resources
- Configure data sources for data synchronization from MySQL
- Configure data sources for data synchronization from PolarDB
- Configure data sources for data synchronization from Oracle
- Add data sources

Context

DataWorks provides a sync solution that can be used to synchronize all data in a database to DataHub in real time. The synchronization solution synchronizes all data in the database to DataHub in batch mode and then synchronizes incremental data in the database to DataHub in real time. You can view the details of the sync solution, the statuses of the nodes generated by the solution, and data updates in the database in real time. This facilitates subsequent data searches, analysis, and development.

Real-time sync solutions that are used to synchronize all data in a database provide the following benefits:

- Synchronizes the full data of a database.

You do not need to create multiple batch data synchronization nodes to synchronize source tables one by one. You can directly create a batch synchronization solution to synchronize some or all of the tables in a database at a time.

- You can configure synchronization rules in a flexible manner.
  - You can configure synchronization rules for different data definition language (DDL) messages based on your business requirements. For example, if you select **Ignore** for a DDL message that is specified in the source and used to drop a table in the destination, the system ignores the message and does not drop the table in the destination when the system receives the message.
  - You can add or remove source tables for a sync solution that is running.
  - You can configure synchronization rules for destination DataHub topics to determine whether to synchronize the incremental data in source tables to destination DataHub topics based on your business requirements. After the incremental data is synchronized, the incremental data can be searched in destination DataHub topics.
- Requires only simple configurations.

  You do not need to perform complex operations, such as creating synchronization nodes, databases, and tables, configuring dependencies for nodes, and configure mappings between sources and destinations. Instead, you need only to configure a batch synchronization solution in a configuration wizard.
- Large amounts of data can be updated in real time. This improves the efficiency of automated O&M.

Scenarios

If you want the system to monitor data updates in business databases in real time, you can use real-time sync solutions to synchronize all data in the databases. This way, upper-layer applications can search for, analyze, and develop data in real time.

Limits

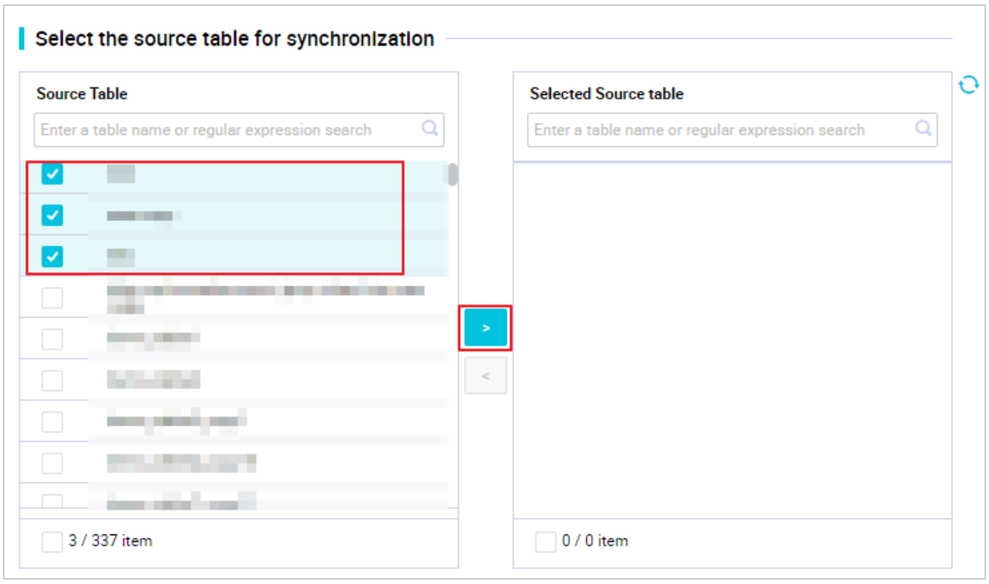
- A real-time sync solution that is used to synchronize all data in a database can synchronize data only from MySQL, PolarDB, or Oracle to DataHub.
- A real-time sync solution that is used to synchronize all data in a database can be run only on exclusive resource groups.

Create a real-time sync solution to synchronize all data in a database

1.
2.
3. On the **Create Data Synchronization Solution** page, click **One-click real-time synchronization to DataHub**.
4.
5. Select a source and configure synchronization rules.
   i. In the **Data Source** section, specify the **Type** and **Data source** parameters.

   > Note
   >
   > A real-time sync solution that is used to synchronize all data in a database can synchronize data only from MySQL, PolarDB, or Oracle to DataHub.

ii. In the **Source Table** section, select the tables whose data you want to synchronize from the **Source Table** list. Then, click the > icon to move the tables to the **Selected Source Table** list.



Select the source table for synchronization

Source Table
Enter a table name or regular expression search
☑ [redacted]
☑ [redacted]
☑ [redacted]
□ [redacted]
□ [redacted]
□ [redacted]
□ [redacted]
□ [redacted]
□ [redacted]
□ 3 / 337 item

Selected Source table
Enter a table name or regular expression search
□ 0 / 0 item

The Source Table list displays all tables in the selected source. You can choose to synchronize data in some or all tables in the source.

iii. In the **Conversion Rule for Table Name** section, click **Add rule** to select a rule.

Supported options include **Conversion Rule for Table Name** and **Rule for Destination Topic**.

- **Conversion Rule for Table Name**: the rule for converting the names of source tables to those of destination topics.
- **Rule for Destination Topic**: the rule for adding prefixes and suffixes to destination topics.

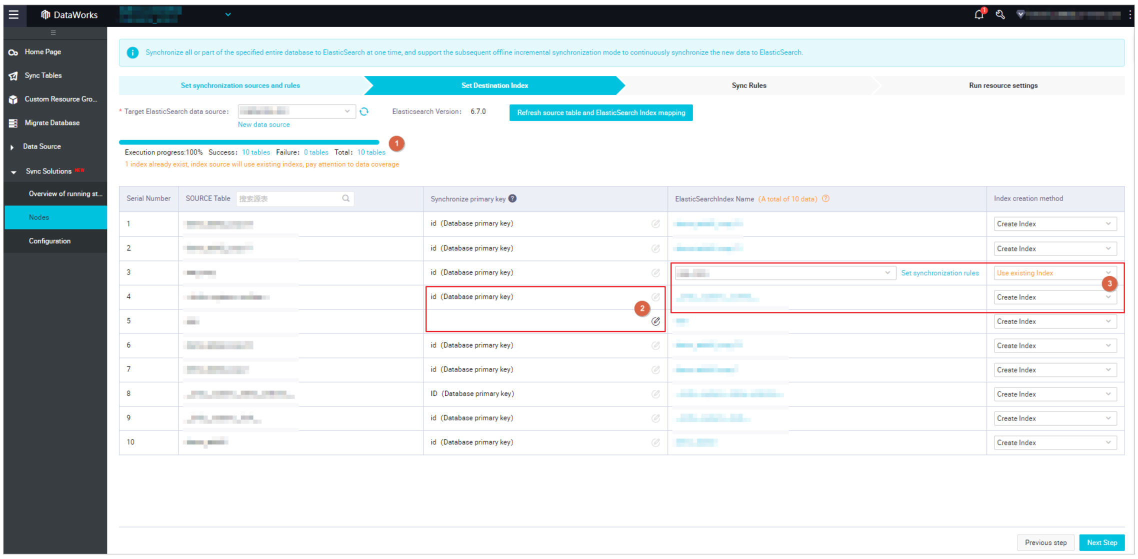
iv. Click **Next Step**.

6. Select a data source as the destination and configure the destination topics.

i. In the **Set Destination Topic** step, select the destination DataHub data source.

ii. Click **Refresh source table and DataHub Topic mapping** to configure the mappings between the source tables and destination DataHub topics.

iii. View the mapping progress, source tables, and mapped destination DataHub topics.



Synchronize all or part of the specified entire database to ElasticSearch at one time, and support the subsequent offline incremental synchronization mode to continuously synchronize the new data to ElasticSearch.

* Target ElasticSearch data source: [redacted]
New data source
Elasticsearch Version: 6.7.0
Refresh source table and ElasticSearch Index mapping

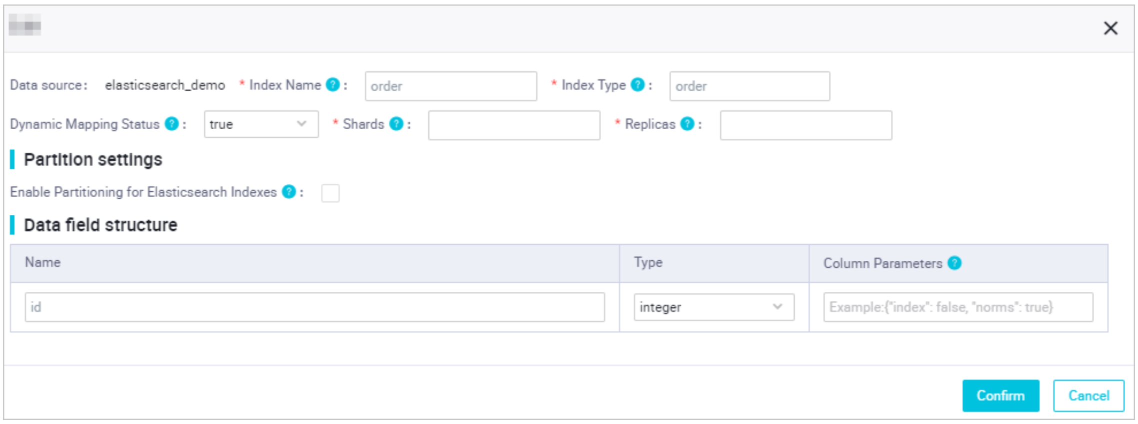
Execution progress:100% Success: 10 tables Failure: 0 tables Total: 10 tables

1 index already exist, index source will use existing indexs, pay attention to data coverage

Serial Number	SOURCE Table 搜索源表	Synchronize primary key	ElasticSearchIndex Name (A total of 10 data)	Index creation method
1	[redacted]	id (Database primary key)	[redacted]	Create Index
2	[redacted]	id (Database primary key)	[redacted]	Create Index
3	[redacted]	id (Database primary key)	[redacted] Set synchronization rules	Use existing Index
4	[redacted]	id (Database primary key)	[redacted]	Create Index
5	[redacted]		[redacted]	Create Index
6	[redacted]	id (Database primary key)	[redacted]	Create Index
7	[redacted]	id (Database primary key)	[redacted]	Create Index
8	[redacted]	ID (Database primary key)	[redacted]	Create Index
9	[redacted]	id (Database primary key)	[redacted]	Create Index
10	[redacted]	id (Database primary key)	[redacted]	Create Index

No.	Description
①	The progress of mapping the source tables to destination DataHub topics. **Note** The mapping may require a long period of time if you synchronize data from a large number of tables.
②	■ If the tables in the source database contain primary keys, the system removes duplicate data based on the primary keys during the synchronization. ■ If the tables in the source database do not contain primary keys, you can click the icon to customize primary keys. You can use one field or a combination of several fields as the primary keys of the tables. This way, the system removes duplicate data based on the primary keys during the synchronization.
③	The methods of creating the destination DataHub topics. ■ If you set **Topic creation method** to **Create Topic** for a destination DataHub topic, the DataHub topic is automatically created. The name of the DataHub topic is displayed in the **DataHub Topic** column. You can click the name of the DataHub topic to modify the configurations of the topic. ■ If you set **Topic creation method** to **Use Existing Topic** for a destination DataHub topic, select the topic that you want to use from the drop-down list in the **DataHub Topic** column.

If you set **Topic creation method** to **Create Topic** for a destination DataHub topic, you can click the name of the DataHub topic to modify the configurations of the topic based on your business requirements.



[redacted]

Data source: elasticsearch_demo * Index Name: order * Index Type: order

Dynamic Mapping Status: true * Shards: * Replicas:

Partition settings

Enable Partitioning for Elasticsearch Indexes: □

Data field structure

Name	Type	Column Parameters
id	integer	Example:{"index": false, "norms": true}

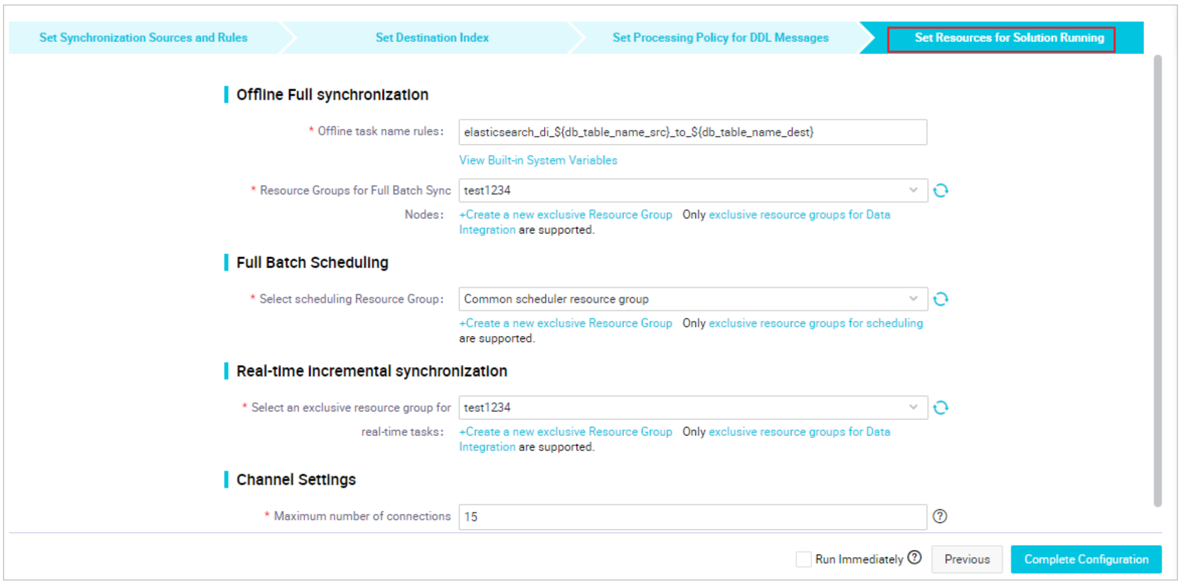
- **Create Topic in Production Environment**: indicates whether to create the topic in the production environment. This option is displayed for a DataWorks workspace in standard mode and is selected by default.
- **Life cycle**: the lifecycle of the topic. Unit: days. Default value: 7.
- **Data field structure**: the fields and their data types in the topic.

> **Note** If you do not change the values of the parameters related to a topic after the topic is created, the system synchronizes data based on the default values of the parameters.

iv. Click **Next**.

7. Configure the resources required by the data sync solution.

   In the **Set Resources for Solution Running** step, set the parameters.



Set Synchronization Sources and Rules
Set Destination Index
Set Processing Policy for DDL Messages
Set Resources for Solution Running

Offline Full synchronization

* Offline task name rules: elasticsearch_di_${db_table_name_src}_to_${db_table_name_dest}

View Built-in System Variables

* Resource Groups for Full Batch Sync Nodes: test1234

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

Full Batch Scheduling

* Select scheduling Resource Group: Common scheduler resource group

+Create a new exclusive Resource Group Only exclusive resource groups for scheduling are supported.

Real-time Incremental synchronization

* Select an exclusive resource group for real-time tasks: test1234

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

Channel Settings

* Maximum number of connections 15

- Offline Full synchronization

Parameter	Description
Offline task name rules	The name of the batch sync node that is used to synchronize the full data of the source. After a data sync solution is created, DataWorks first generates a batch sync node to synchronize full data, and then generates real-time sync nodes to synchronize incremental data.
Resource Groups for Full Batch Sync Nodes	The exclusive resource group for Data Integration that is used to run the batch sync node. Only exclusive resource groups for Data Integration can be used to run sync solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Full Batch Scheduling

Parameter	Description
Select scheduling Resource Group	The resource group for scheduling that is used to run the nodes. Only exclusive resource groups for Data Integration can be used to run sync solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Real-time Incremental synchronization

Parameter	Description

Parameter	Description
Select an exclusive resource group for real-time tasks	The exclusive resource group that is used to run the real-time sync nodes. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- **Channel Settings**

Parameter	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: 15.

8. Click **Complete Configuration**. The real-time sync solution used to synchronize all data in a database is created.

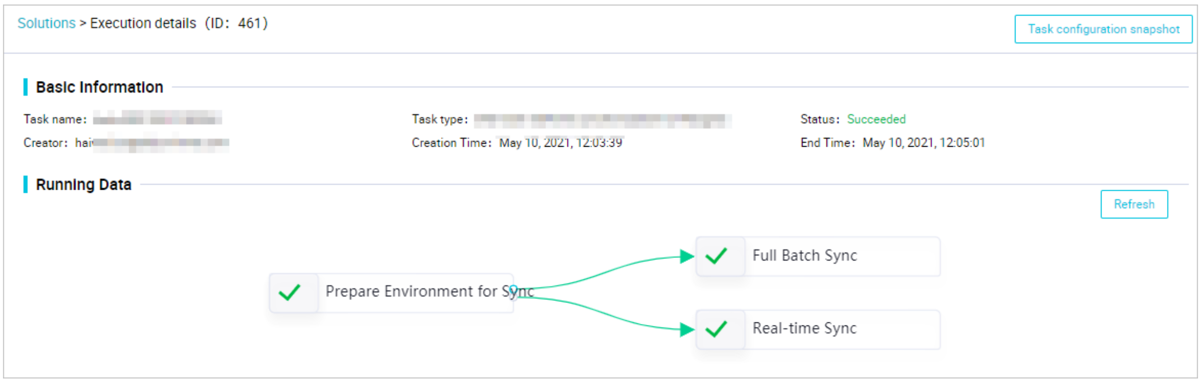
> **Note**
> - The mapping may require a long period of time if you synchronize data from a large number of tables.
> - Only exclusive resource groups for data integration can be used to run real-time sync nodes. For more information, see Create and use an exclusive resource group for Data Integration.

Run the real-time sync solution

On the **Tasks** page, find the newly created data sync solution and choose More > **Submit and Run** in the Operation column to run the data sync solution.

View the statuses and running results of the sync nodes

- On the **Tasks** page, find the solution that is run and click **Execution details** in the Operation column. Then, you can view the execution details of all nodes generated by the sync solution.



Solutions > Execution details (ID: 461)

Task configuration snapshot

Basic Information

Task name: [redacted]

Task type: [redacted]

Status: Succeeded

Creator: hai[redacted]

Creation Time: May 10, 2021, 12:03:39

End Time: May 10, 2021, 12:05:01

Running Data

Refresh

- Find a node whose execution details you want to view and click **Execution details** in the Status column. Then, you can click the link provided in the dialog box that appears to go to the DataStudio page.

Manage the real-time sync solution

- View the configurations of the sync solution.

  On the **Tasks** page, find the newly created sync solution and choose **More > View Configuration**. Then, you can view the configurations of the sync solution.

- Modify the sync solution.

On the **Tasks** page, find the newly created sync solution and choose **More > Modify Configuration**. Then, you can modify the configurations of the sync solution.

For a sync solution that is successfully run, you can choose **More > Modify Configuration** to add or remove source tables. Procedure:

In the **Source Table** section of the **Set Synchronization Sources and Rules** step, add or remove source tables for the sync solution. Then, save the modification and run the sync solution.

- Change the priority for the batch synchronization solution

  Find the newly created batch synchronization solution and choose **More > Change Priority** in the Operation column. In the **Change Priority** dialog box, enter the desired priority and click **Confirm**. You can set the priority to an integer from 1 to 8. A larger value indicates a higher priority.

  > ⓘ **Note** If multiple batch synchronization solutions have the same priority, the system runs them based on the order they are committed.

- Delete the batch synchronization solution.

  Find the batch synchronization solution that you want to delete and choose **More > Delete** in the Operation column. In the Delete message, click **OK**.

  > ⓘ **Note** After you click OK, only the configuration record of the batch synchronization solution is deleted. The synchronization nodes generated by the solution and data tables generated by the synchronization nodes are not affected.

4.4. Synchronize data to Elasticsearch

4.4.1. Plan and configure resources

When you use DataWorks to synchronize data, you can use only exclusive resource groups for Data Integration to run Data Integration nodes. In addition, you can select a shared or exclusive resource group for scheduling based on your business requirements. This topic describes the resources that are used for data synchronization and how to configure the resources.

Context

- Resource planning and preparation

  When you synchronize data, Data Integration nodes run on resources in resource groups for Data Integration and resource groups for scheduling. You can use only exclusive resource groups for Data Integration. Before you synchronize data, you must purchase an exclusive resource group for Data Integration and add the exclusive resource group to your DataWorks workspace.

  For more information about exclusive resource groups for Data Integration, see Exclusive resources for Data Integration.

- Network connections

  An exclusive resource group for Data Integration is essentially a group of ECS instances. After you purchase such an exclusive resource group, it is isolated from other services. You must associate the resource group with a virtual private cloud (VPC) to ensure network connectivity between the resource group and data sources during subsequent data synchronization.

What's next

After you plan and configure resources, you can configure data sources. You must connect the exclusive resource group for Data Integration to the source and destination. You must also create an account and grant the required permissions to the account. This account is used to access the source and destination. The preceding operations help create a data synchronization node. For more information about how to configure data sources, see Configure data sources for data synchronization from MySQL and Configure data sources for data synchronization from PolarDB.

4.4.2. Configure data sources for data synchronization from MySQL

Before you use DataWorks to synchronize data from a MySQL data source to an Elasticsearch cluster, you can refer to the operations in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions.

Prerequisites

Before you configure the data sources, make sure that the following operations are performed:

- Prepare data sources: A MySQL database and a destination Elasticsearch cluster are prepared.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.
- Execute the following statement to check the version of your MySQL database. You can use a `MySQL 5.X` or `MySQL 8.X` database.

```
select version();
```

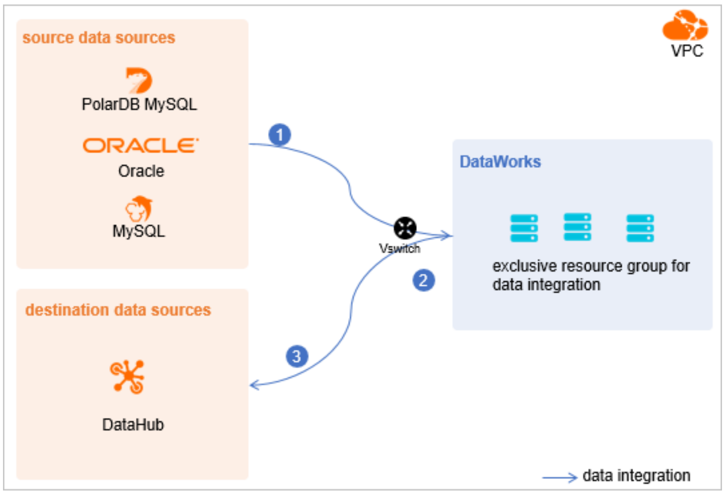
Note Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.x` or `V8.x`. PolarDB for MySQL is not supported. If the MySQL version of your ApsaraDB RDS for MySQL database is not `V5.x` or `V8.x`, use an ApsaraDB RDS for MySQL database whose MySQL version is `V5.x` or `V8.x`. Otherwise, the data synchronization node fails to run.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

  If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the vSwitch that is bound to the exclusive resource group for Data Integration during network configuration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can access the data sources.



- Create an account and grant permissions the account

You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source data source is a MySQL database, you must enable the binary logging feature. Binary logs record changes to all table schemas and modifications on table data. You can execute statements such as CREATE and ALTER to perform operations on table schemas. You can execute statements such as INSERT, UPDATE, and DELETE to perform operations on table data. You can use binary logs to view the change history of the database, back up incremental data and restore data in the database, and replicate data from the primary database to secondary databases.

  Formats of binary logs:

  - Statement: SQL statement-based replication. Binary logs in this format record the SQL statements that are executed to modify data entries.
  - Row: row-based replication. Binary logs in this format record only modification details about data entries in rows.
  - Mixed: replication in mixed mode. This mode combines the statement and row formats. In most cases, binary logs in the statement format are used to record the SQL statements that are executed to modify data entries, such as functions. If data replication from the primary database to secondary databases cannot be implemented by using binary logs in this format, switch to the row format. MySQL determines which format to use based on each SQL statement that is executed.

Limits

Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.X` or `V8.X`. PolarDB for MySQL is not supported.

Procedure

1. Configure a whitelist for the MySQL database.

   Add the CIDR block of the VPC in which the exclusive resource group resides to the whitelist of the MySQL database.

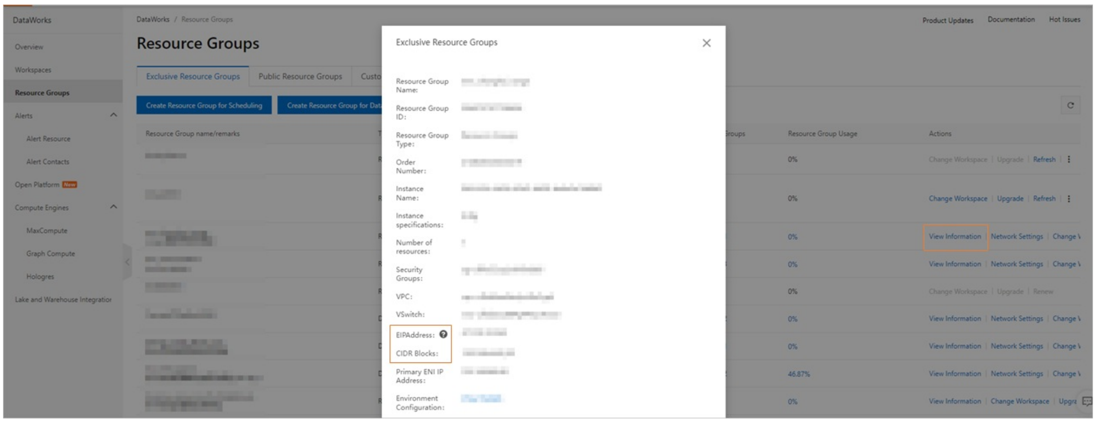
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

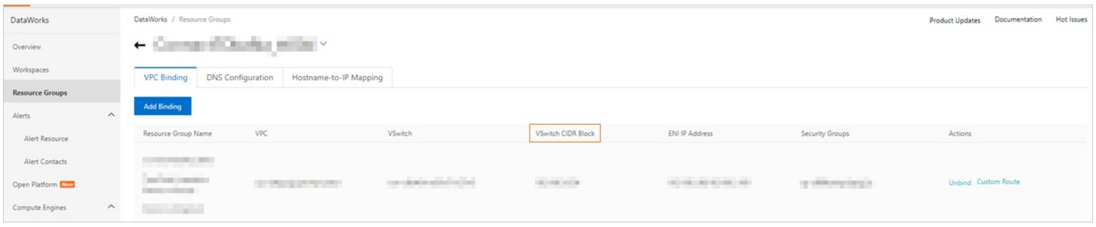
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR block of the exclusive resource group to the whitelist of the MySQL database.

2. Create an account and grant permissions to the account.

You must create an account to log on to the MySQL database. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

i. Create an account.

For more information, see Create an account to access a MySQL database.

ii. Grant permissions to the account.

You can run the following command to grant permissions to the account. Alternatively, you can grant the `SUPER` permission to the account. Replace `Account for data synchronization` with the created account when you execute the specific statement.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Password'; // Create an account that is used to synchronize data and set a password so that you can use the account and password to access the database from a host. % indicates a host.
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%'; // Grant the SELECT, REPLICATION SLAVE, and REPLICATION CLIENT permissions to the account.
```

`*.*` indicates that the synchronization account is granted the preceding permissions on all tables in all databases. You can also grant the preceding permissions on specified tables in the specified database to the synchronization account. For example, to grant the account the preceding permissions on the *user* table in the *test* database, execute the following statement: `GRANT SELECT, REPLICATION CLIENT ON test.user TO 'Account for data synchronization'@'%';` .

> **Note** The `REPLICATION SLAVE` permission is a global permission. You cannot grant this permission on specified tables in the specified database to the synchronization account.

3. Enable the binary logging feature for the MySQL database.

   Perform the following steps to check whether the binary logging feature is enabled and to query the format of binary logs:

   - Execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_bin";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled.

   - If you use a secondary database to synchronize data, execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_slave_updates";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled for the secondary database.

   If the returned result is different from the preceding result, follow the instructions described in the *MySQL documentation* to enable the binary logging feature.

   Execute the following statement to view the format of binary logs:

   ```
   show variables like "binlog_format";
   ```

   Returned result:

   - *ROW*: The format of binary logs is row.
   - *STATEMENT*: The format of binary logs is statement.
   - *MIXED*: The format of binary logs is mixed.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add data sources.

4.4.3. Configure data sources for data synchronization from PolarDB

Before you synchronize data from a PolarDB data source to Elasticsearch, you can refer to the operations in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions.

Prerequisites

Before you configure the data sources, make sure that the following operations are performed:

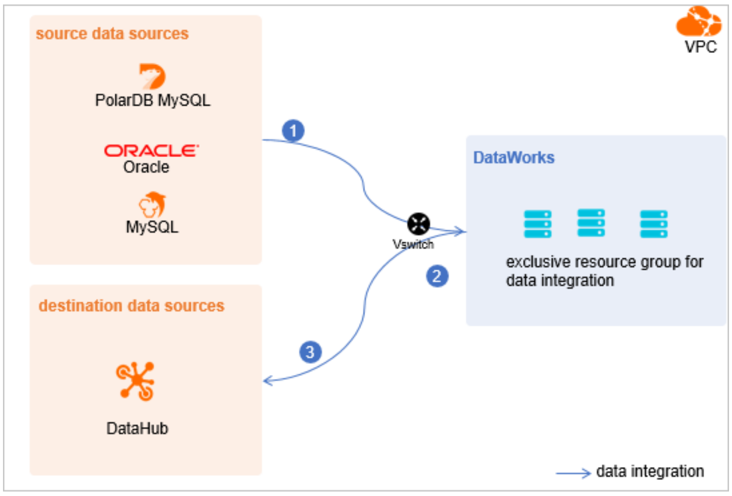
- Prepare data sources: A source PolarDB for MySQL cluster and a destination Elasticsearch cluster are created. In this topic, a PolarDB for MySQL cluster is used as the source.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

  If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the vSwitch that is bound to the exclusive resource group for Data Integration during network configuration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source is a PolarDB for MySQL cluster, you must enable the binary logging feature for the cluster. PolarDB for MySQL is fully compatible with MySQL and uses high-level physical logs to replace binary logs. To facilitate the integration between PolarDB and the MySQL ecosystem, you can enable the binary logging feature for PolarDB clusters.

Limits

- Only PolarDB for MySQL clusters can be used as sources in data synchronization solutions. Other types of PolarDB data sources are not supported. In this topic, PolarDB indicates PolarDB for MySQL data sources.
- Only data stored on the primary node of a PolarDB for MySQL cluster can be synchronized.

Procedure

1. Configure a whitelist for the PolarDB for MySQL cluster.

   To add the CIDR block of the VPC where the exclusive resource group for Data Integration resides to a whitelist of the PolarDB for MySQL cluster, perform the following steps:

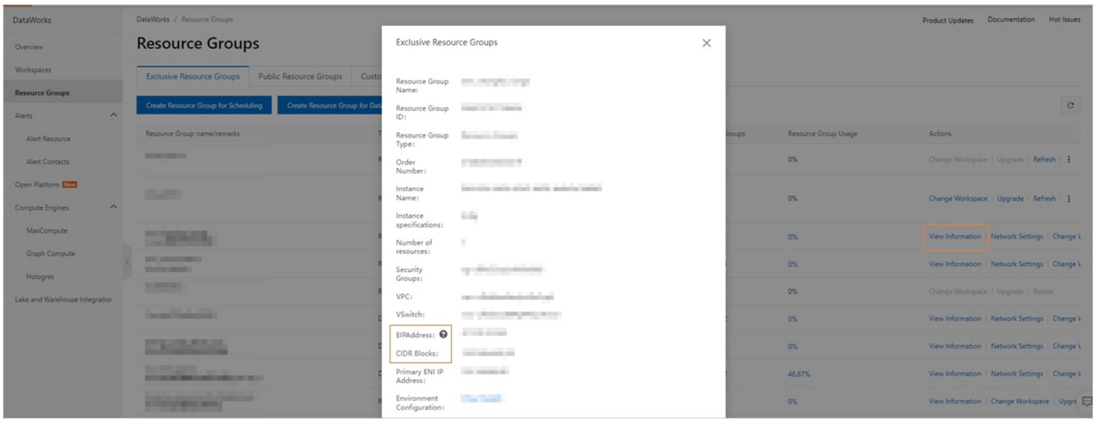
   i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

      a. Log on to the DataWorks console.

      b. In the left-side navigation pane, click **Resource Groups**.

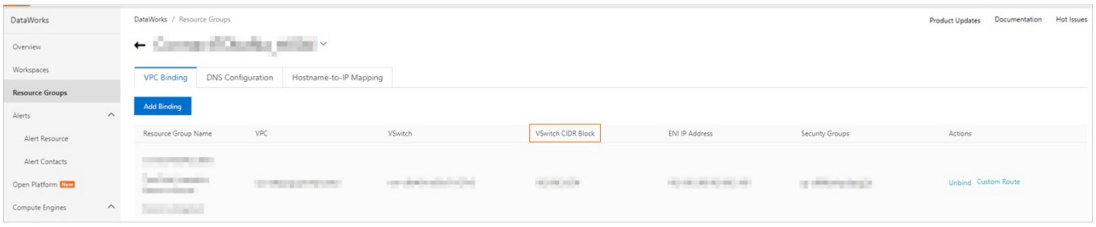
      c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

      d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.

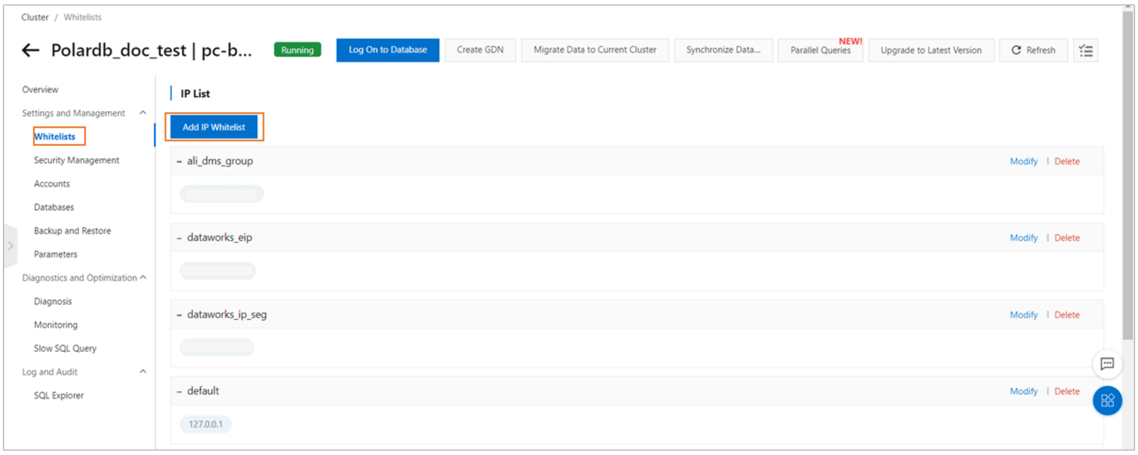


      e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

      f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



   ii. Add the EIP and CIDR blocks recorded in the preceding steps to the whitelist of the PolarDB for MySQL cluster.



   For more information, see Configure an IP whitelist.

2. Create an account and grant the required permissions to the account.

   You must create an account to log on to the database of the PolarDB for MySQL cluster. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

   i. Create an account.

      For more information, see Create a database account.

   ii. Grant the required permissions to the account.

      You can run the following command to grant the required permissions to the account, or you can directly assign the `SUPER` role to the account.

      ```
      -- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Account for data synchronization';
      GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%';
      ```

3. Enable the binary logging feature for the PolarDB for MySQL cluster.

   For more information, see Enable binary logging.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add data sources.

4.4.4. Add data sources

Before you configure a real-time data synchronization node to synchronize data from a data source to an Elasticsearch data source, you must add both data sources to DataWorks for subsequent source and destination configurations.

Prerequisites

Before you add data sources, make sure that the following operations are performed:

- Prepare data sources: A source data source and a destination data source are created.
- Create and grant permission to an account: An account that is used to access data sources is created.

Precautions

DataWorks provides workspaces in basic mode and standard mode. A workspace in basic mode does not isolate the development environment from the production environment. A workspace in standard mode isolates the development environment from the production environment.

If you use a workspace in standard mode, you must separately add data sources to the development environment and production environment.

Add a source MySQL data source

To add a source MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source MySQL data source, see Add a MySQL data source.

Add a source PolarDB for MySQL data source

To add a source PolarDB for MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source PolarDB for MySQL data source, see Add a PolarDB data source.

If the source PolarDB for MySQL data source that you want to add fails the connectivity test, follow the instructions described in What do I do if the PolarDB data source cannot be connected? to handle the exception.

Add the destination (Elasticsearch)

For more information, see Add an Elasticsearch data source.

What's next

After you add data sources, you can create and run a data sync node to synchronize data from the source data source to the destination data source.

For more information, see Configure and view a batch synchronization solution used to synchronize all data in a database or Configure and view a real-time sync solution used to synchronize all data in a database.

4.4.5. Configure and view a batch synchronization solution used to synchronize all data in a database

After you configure data sources, network environments, and resource groups, you can create and run a batch synchronization solution to synchronize all data in a database. This topic describes how to create a batch synchronization solution to synchronize data in some or all tables in a database to Elasticsearch. This topic also describes how to view the statuses of the nodes generated by the batch synchronization solution.

Prerequisites

Before you create a data synchronization solution, make sure that the following operations are performed:

- Plan and configure resources
- Configure data sources for data synchronization from MySQL
- Configure data sources for data synchronization from PolarDB
- Add data sources

Context

In most cases, the real-time data of enterprises are stored in big data engines, and a large volume of non-structured log data may be generated for the real-time data. You can use the hot-warm architecture that is provided by Elasticsearch in a fully managed manner to store the log data and offline data of enterprises. DataWorks provides batch synchronization solutions that can be used to synchronize all data in a database to Elasticsearch based on the architecture. You can view the details of the solution and the statuses of the nodes generated by the solution. This makes automated operations and maintenance (O&M) and management more efficient.

You can use a batch synchronization solution to synchronize the full or incremental data in your business database to Elasticsearch. Then, the data can be searched, analyzed, and developed in Elasticsearch. A batch synchronization solution used to synchronize all data in a database has the following benefits:

- Synchronizes the full data of a database.

  You do not need to create multiple batch data synchronization nodes to synchronize source tables one by one. You can directly create a batch synchronization solution to synchronize some or all of the tables in a database at a time.

- Supports various data synchronization methods.

  You can use one of the following methods to synchronize data: full data synchronization, incremental data synchronization, and a combination of full and incremental data synchronization. In addition, you can configure properties for your batch synchronization solution.

- Requires only simple configurations.

  You do not need to perform complex operations, such as creating synchronization nodes, databases, and tables, configuring dependencies for nodes, and configure mappings between sources and destinations. Instead, you need only to configure a batch synchronization solution in a configuration wizard.

- Reduces costs and improves O&M efficiency.

Limits

- You can use a batch synchronization solution to synchronize all data only in a MySQL, SQL Server, or PolarDB database to Elasticsearch.
- A batch synchronization solution used to synchronize all data in a database can be run only on resources in exclusive resource groups for Data Integration.

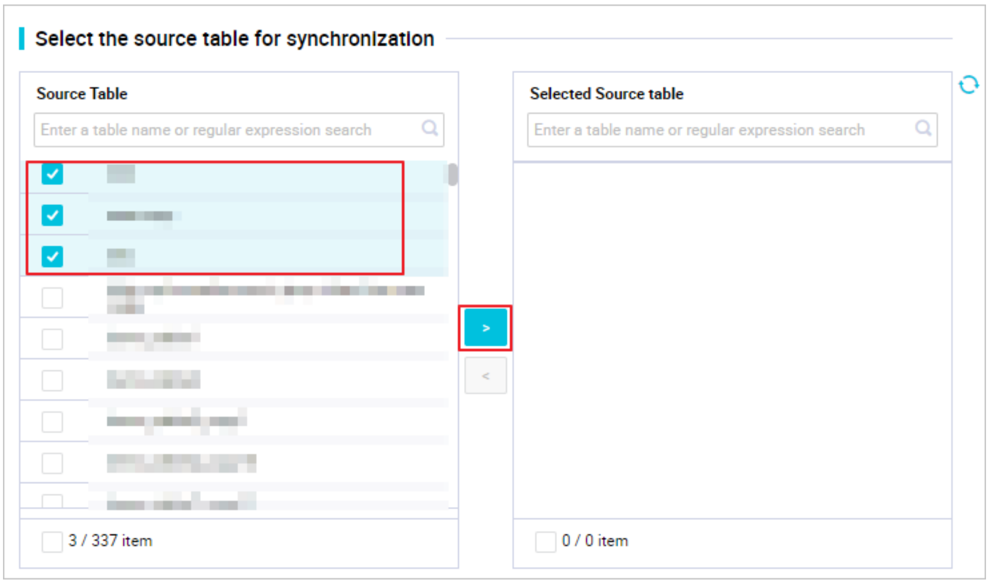
Create a batch synchronization solution to synchronize all data in a database

1.
2.
3. In the **New synchronization solution** dialog box, click **One-click batch synchronization to Elasticsearch**.
4.
5. Select a data source as the source and configure synchronization rules.

i. In the **Data source** section, specify the **Type** and **Data source** parameters.

> **Note** You can select MySQL, SQL Server, or PolarDB as the source.

ii. In the **Source Table** section, select the tables whose data you want to synchronize from the **Source Table** list. Then, click the > icon to move the tables to the **Selected Source Table** list.


Select the source table for synchronization
Source Table
Enter a table name or regular expression search
Selected Source table
Enter a table name or regular expression search
>
<
3 / 337 item
0 / 0 item

The Source Table list displays all tables in the selected source. You can choose to synchronize data in some or all tables in the source.

> **Notice** If a selected table has no primary key, you must customize a primary key when you map the table to a destination Elasticsearch index. This primary key is used to remove duplicate data during synchronization. For example, you can use one field or a combination of several fields as the primary key of the table. For more information, see Step 6 in this topic.

iii. In the **Conversion Rule for Table Name** section, click **Add rule** to select a rule.

Supported options include **Conversion Rule for Table Name** and **Rule for Destination Index Name**.

- **Conversion Rule for Table Name**: the rule for converting the names of source tables to those of destination Elasticsearch indexes.
- **Rule for Destination Index Name**: the rule for adding a prefix and a suffix to the converted names of destination Elasticsearch indexes.

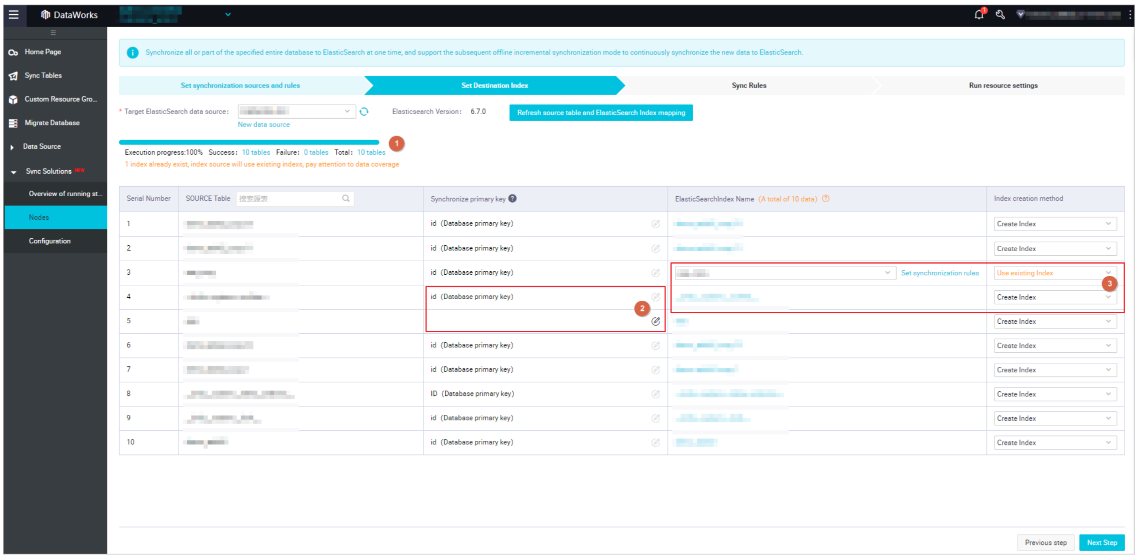
iv. Click **Next Step**.

6. Select a destination cluster and configure destination Elasticsearch indexes.

i. In the **Set Destination Index** step, specify **Destination**.

ii. Click **Refresh source table and Elasticsearch Index mapping** to configure the mappings between the source tables and destination Elasticsearch indexes.

iii. View the mapping progress, source tables, and mapped destination Elasticsearch indexes.



Synchronize all or part of the specified entire database to ElasticSearch at one time, and support the subsequent offline incremental synchronization mode to continuously synchronize the new data to ElasticSearch.

* Target ElasticSearch data source: [redacted] Elasticsearch Version: 6.7.0

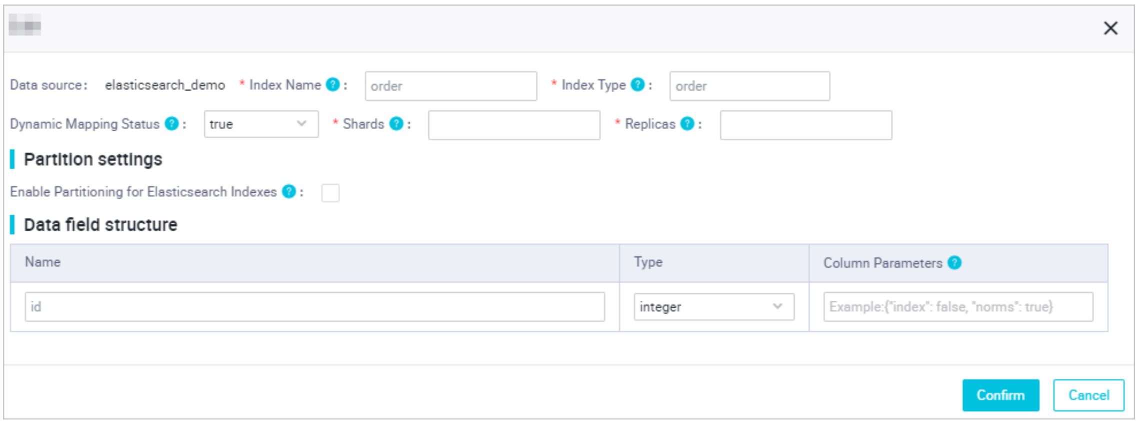
Execution progress:100% Success: 10 tables Failure: 0 tables Total: 10 tables

1 index already exist, index source will use existing indexs, pay attention to data coverage

Serial Number	SOURCE Table 搜索源表	Synchronize primary key	ElasticSearchIndex Name (A total of 10 data)	Index creation method
1	[redacted]	id (Database primary key)	[redacted]	Create Index
2	[redacted]	id (Database primary key)	[redacted]	Create Index
3	[redacted]	id (Database primary key)	[redacted] Set synchronization rules	Use existing Index
4	[redacted]	id (Database primary key)	[redacted]	Create Index
5	[redacted]		[redacted]	Create Index
6	[redacted]	id (Database primary key)	[redacted]	Create Index
7	[redacted]	id (Database primary key)	[redacted]	Create Index
8	[redacted]	ID (Database primary key)	[redacted]	Create Index
9	[redacted]	id (Database primary key)	[redacted]	Create Index
10	[redacted]	id (Database primary key)	[redacted]	Create Index

No.	Description
1	The progress of mapping the source tables to destination Elasticsearch indexes. ⓘ **Note** The mapping may require a long time if you want to synchronize data from a large number of tables.
2	■ If the tables in the source database contain primary keys, the system removes duplicate data based on the primary keys during the synchronization. ■ If the tables in the source database do not contain primary keys, you can click the ✎ icon to customize primary keys. You can use one field or a combination of several fields as the primary keys of the tables. This way, the system removes duplicate data based on the primary keys during the synchronization. ⓘ **Note** In the following cases, you must configure the primary keys: ■ You use an **incremental data synchronization** method to synchronize data. ■ You use a **full data synchronization** method to synchronize data and set **Write Policy** to **Update**. For more information about synchronization methods, see the synchronization methods described in Step 7 in this topic.
3	The method used to create an index. Valid values: ■ **Create Index**: If you select this method, the name of the Elasticsearch index that is automatically created appears in the **Elasticsearch Index Name** column. You can click the name of the index to change the values of the parameters related to the index. ■ **Use Existing Index**: If you select this method, select the name of the desired index from the drop-down list in the **Elasticsearch Index Name** column. Then, you can click **View Field Mapping** to view the mappings between the source tables and destination Elasticsearch indexes.

If you set the **Index creation method** parameter to **Create Index** , you can click the Elasticsearch index name that appears in the Elasticsearch Index Name column to change the values of the parameters related to the destination Elasticsearch index based on your business requirements.



Data source: elasticsearch_demo * Index Name: order * Index Type: order

Dynamic Mapping Status: true * Shards: * Replicas:

Partition settings

Enable Partitioning for Elasticsearch Indexes: □

Data field structure

Name	Type	Column Parameters
id	integer	Example:{"index": false, "norms": true}

- **Dynamic Mapping Status**: specifies whether to dynamically synchronize new fields in the source tables to the destination Elasticsearch indexes during synchronization. Valid values:
  - **true**: If the system detects that the source tables contain new fields, the system synchronizes the fields to the mapped destination Elasticsearch indexes, and the fields can be searched in the indexes after synchronization. Default value: true.
  - **false**: If the system detects that the source tables contain new fields, the system synchronizes the fields to the mapped destination Elasticsearch indexes, but the fields cannot be searched in the indexes after synchronization.
  - **strict**: If the system detects that the source tables contain new fields, the system does not synchronize the fields to the mapped destination Elasticsearch indexes, and an error is reported. You can view the details of the error in the node logs.

  For more information about dynamic mappings, see the description of the dynamic parameter for open source Elasticsearch.
- **Shards** and **Replicas**: the number of primary shards for the destination Elasticsearch index and the number of replica shards for each primary shard. The shards are distributed on different nodes in an Elasticsearch cluster to support distributed searches. This improves the query efficiency of Elasticsearch. For more information, see Terms.

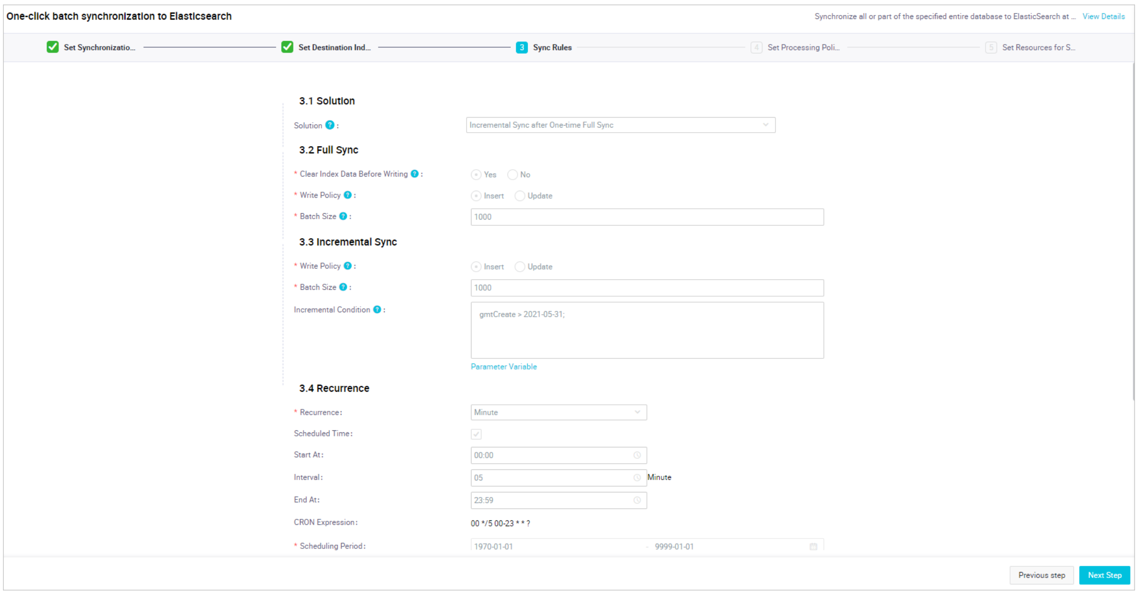
  > Note The values of the **Shards** and **Replicas** parameters cannot be changed after you specify them and the real-time sync solution starts to run. The default values of the Shards and Replicas parameters are 7.
- **Partition settings**: You can use a column in a source table as a partition key column. This parameter must be used together with the **Shards** and **Replicas** parameters. By default, the Enable Partitioning for Elasticsearch Indexes check box is not selected.
- **Data field structure**: This section allows you to configure the types and extended attributes of the fields in the mapped destination Elasticsearch indexes. For more information, see Field data types in open source Elasticsearch.

> Note If you do not change the values of the parameters related to the destination Elasticsearch indexes after the indexes are created, the system synchronizes data based on the default values of the parameters.

iv. Click **Next Step**.

7. Configure synchronization rules.

i. In the **Sync Rules** step, select a synchronization method.



One-click batch synchronization to Elasticsearch

Synchronize all or part of the specified entire database to ElasticSearch at ... View Details

✓ Set Synchronizatio... — ✓ Set Destination Ind... — 3 Sync Rules — 4 Set Processing Poli... — 5 Set Resources for S...

3.1 Solution

Solution: Incremental Sync after One-time Full Sync

3.2 Full Sync

* Clear Index Data Before Writing: ● Yes ○ No

* Write Policy: ● Insert ○ Update

* Batch Size: 1000

3.3 Incremental Sync

* Write Policy: ● Insert ○ Update

* Batch Size: 1000

Incremental Condition: gmtCreate > 2021-05-31;

Parameter Variable

3.4 Recurrence

* Recurrence: Minute

Scheduled Time: ☑

Start At: 00:00

Interval: 05 Minute

End At: 23:59

CRON Expression: 00 */5 00-23 * * ?

* Scheduling Period: 1970-01-01 - 9999-01-01

The following table describes the synchronization methods.

Method	Description
Only One-time Full Sync	If you use this method, you need only to perform synchronization operations once to synchronize all data in the source to Elasticsearch.
Only One-time Incremental Sync	If you use this method, you need only to perform synchronization operations once to synchronize incremental data in the source to Elasticsearch based on the specified filter conditions.
Periodic Full Sync	If you use this method, you must configure a scheduling cycle for the batch synchronization solution. Then, the system synchronizes all data in the source to Elasticsearch each time the system runs the solution based on the specified scheduling cycle.
Periodic Incremental Sync	If you use this method, the system synchronizes only incremental data in the source to Elasticsearch each time the system runs the solution based on the specified filter conditions and scheduling cycle.
Incremental Sync after One-time Full Sync	If you use this method, the system first synchronizes all data to Elasticsearch. Then, the system synchronizes only incremental data in the source to Elasticsearch each time the system runs the solution based on the specified filter conditions and scheduling cycle.

ii. Configure parameters for the selected synchronization method.

The parameters that you need to specify in the **Full Sync**, **Incremental Sync**, and **Recurrence** sections vary based on the synchronization method you selected. The following tables describe the parameters.

- Full Sync

  The parameters in this section are required only if you set **Solution** to **Only One-time Full Sync**, **Periodic Full Sync**, or **Incremental Sync after One-time Full Sync**.

Parameter	Description
Clear Index Data Before Writing	Valid values: - **Yes**: The original data in the destination Elasticsearch indexes is deleted before data in the source is written to the indexes. - **No**: The original data in the destination Elasticsearch indexes is retained before data in the source is written to the indexes. **Notice** If you set this parameter to **Yes**, all the original data in the destination Elasticsearch indexes is deleted before data in the source is written to the indexes. Exercise caution when you set this parameter.
Write Policy	Valid values: - **Insert**: The system inserts data to the destination Elasticsearch indexes during data synchronization. This is the default value of this parameter. - **Update**: If the primary key field of a source table already exists in a destination Elasticsearch index, the system first deletes a document in the destination Elasticsearch index and then inserts data to the index. Otherwise, the system directly inserts data to the destination Elasticsearch index.
Batch Size	The number of data records that can be written to Elasticsearch at a time. Default value: **1000**. You can set this parameter to an appropriate value based on actual network conditions and the data volume that you want to synchronize. This can reduce network overheads.

- Incremental Sync

  The parameters in this section are required only if you set **Solution** to **Only One-time Incremental Sync**, **Periodic Incremental Sync**, or **Incremental Sync after One-time Full Sync**.

Parameter	Description
Write Policy	Valid values: - **Insert**: The system inserts data to the destination Elasticsearch indexes during data synchronization. This is the default value of this parameter. - **Update**: If the primary key field of a source table already exists in a destination Elasticsearch index, the system first deletes a document in the destination Elasticsearch index and then inserts data to the index. Otherwise, the system directly inserts data to the destination Elasticsearch index.
Batch Size	The number of data records that can be written to Elasticsearch at a time. Default value: **1000**. You can set this parameter to an appropriate value based on actual network conditions and the data volume that you want to synchronize. This can reduce network overheads.
Incremental Condition	The filter conditions that are used to filter data in the source to synchronize only incremental data. You can configure filter conditions based on descriptions in Overview of scheduling parameters.

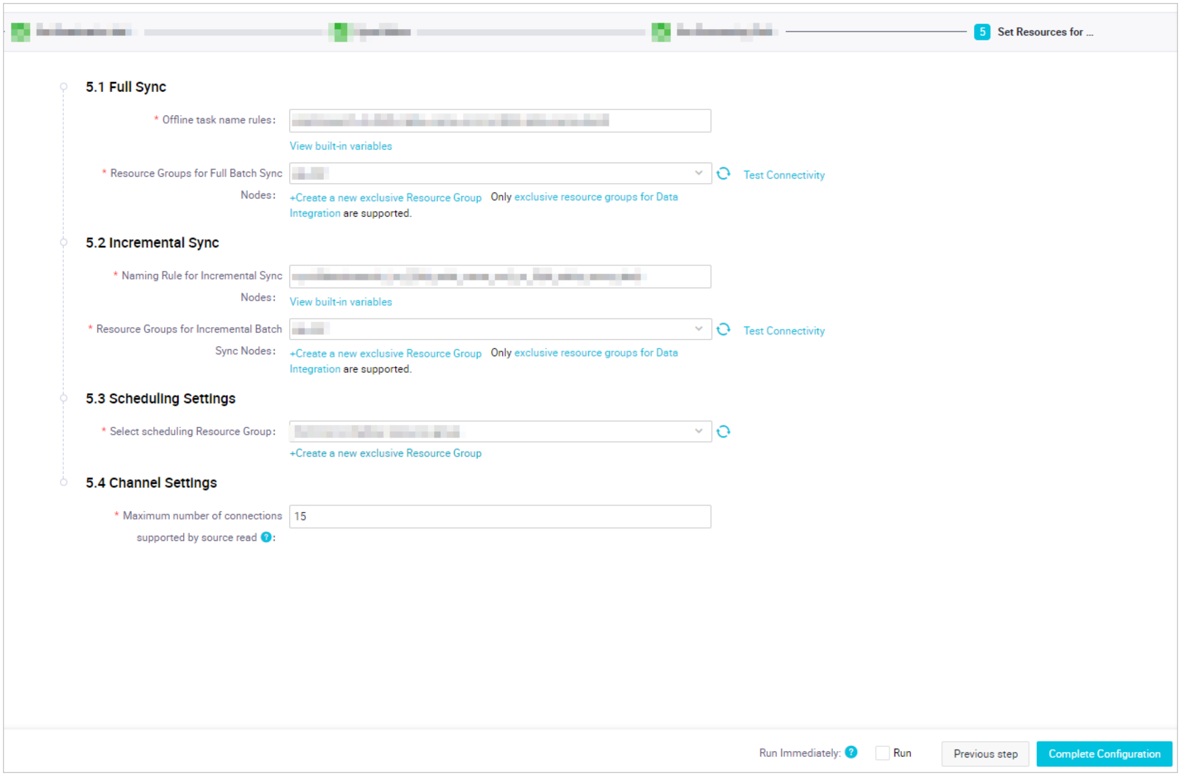
- **Recurrence**

Parameter	Description
Recurrence	The scheduling cycle of the batch synchronization solution. Valid values: **Minute**, **Hour**, **Daily**, **Weekly**, and **Monthly**. For more information about how to configure a scheduling cycle, see Configure time properties.
Scheduling Period	The batch synchronization solution is run only within the scheduling period that you specified.
Pausing Scheduling	If you select Pausing Scheduling, the batch synchronization solution is paused. In this case, the solution starts to run based on the scheduling cycle until you cancel the pausing. You can select this check box if you do not need to run the solution for a period of time.
Rerun	Valid values: - **Allow Regardless of Running Status**: You can set Rerun to this value if the batch synchronization solution can be rerun multiple times and the reruns do not affect synchronization results. - **Disallow Regardless of Running Status**: You can set Rerun to this value if synchronization results can be affected regardless of whether the running of the batch synchronization solution is successful or fails. If you set Rerun to this value, the system does not automatically rerun the synchronization solution after the system recovers from an exception.

iii. Click **Next Step**.

8. Configure the resources required for the synchronization solution.

   In the **Set Resources for Solution Running** step, configure the parameters.



5 Set Resources for ...

5.1 Full Sync

* Offline task name rules: [redacted]

View built-in variables

* Resource Groups for Full Batch Sync Nodes: [redacted] Test Connectivity

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

5.2 Incremental Sync

* Naming Rule for Incremental Sync Nodes: [redacted]

View built-in variables

* Resource Groups for Incremental Batch Sync Nodes: [redacted] Test Connectivity

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

5.3 Scheduling Settings

* Select scheduling Resource Group: [redacted]

+Create a new exclusive Resource Group

5.4 Channel Settings

* Maximum number of connections supported by source read ?: 15

- ○ **Full Sync**

  The parameters in this section are required only if you set **Solution** to **Only One-time Full Sync**, **Periodic Full Sync**, or **Incremental Sync after One-time Full Sync** in the **Sync Rules** step.

Parameter	Description
Offline task name rules	The name of the batch synchronization node that is used to synchronize the full data of the source. After the synchronization solution is created, DataWorks generates a batch node to synchronize the full data of the source.
Resource Group for Full Batch Sync Nodes	Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- **Incremental Sync**

  The parameters in this section are required only if you set **Solution** to **Only One-time Incremental Sync**, **Periodic Incremental Sync**, or **Incremental Sync after One-time Full Sync in the Sync Rules** step.

Parameter	Description
Naming Rule for Incremental Sync Nodes	The name of the batch synchronization node that is used to synchronize the incremental data of the source. After the synchronization solution is created, DataWorks generates a batch synchronization node to synchronize the incremental data of the source.
Resource Group for Incremental Batch Sync Nodes	Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- **Scheduling Settings**

Parameter	Description
Select scheduling Resource Group	The resource group for scheduling that is used to run the nodes generated by the batch synchronization solution. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: **15**.

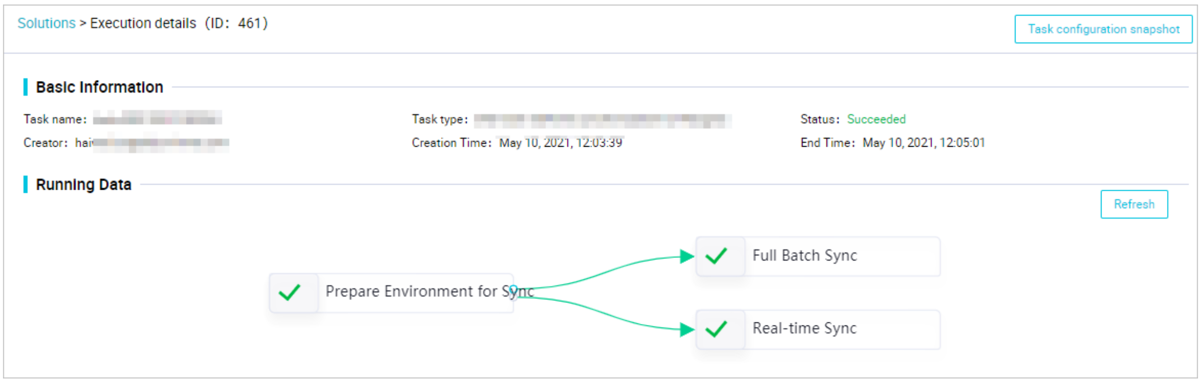
9. Click **Complete configuration**. The batch synchronization solution used to synchronize all data in a database is created.

Run the batch synchronization solution

On the **Tasks** page, find the created data synchronization solution and click **Submit and Run** in the Operation column to run the solution.

View the statuses and results of the synchronization nodes

- On the **Tasks** page, find the solution that is run and click **Execution details** in the Operation column. Then, you can view the details of all nodes generated by the batch synchronization solution.



Solutions > Execution details (ID: 461)

Task configuration snapshot

Basic Information

Task name: [redacted]

Creator: hai[redacted]

Task type: [redacted]

Creation Time: May 10, 2021, 12:03:39

Status: Succeeded

End Time: May 10, 2021, 12:05:01

Running Data

Refresh

- Find a node whose details you want to view and click **Execution details** in the Status column. In the dialog box that appears, click the provided link to go to the DataStudio page.

Manage the real-time synchronization solution

- View or edit the data synchronization solution.

  On the **Tasks** page, find the newly created synchronization solution and choose **More** > **View Setting** or choose **More** > **Modify Configuration** in the Operation column. Then, you can view or modify the configurations of the batch synchronization solution.

  > **Note** You can choose **More** > **Modify Configuration** in the Operation column that corresponds to a batch synchronization solution in the **Not Running** state to edit the batch synchronization solution. If you click Modify Configuration in the Operation column that corresponds to a batch synchronization solution in another state, you can only view information about the solution.

- Change the priority for the batch synchronization solution

  Find the newly created batch synchronization solution and choose **More** > **Change Priority** in the Operation column. In the **Change Priority** dialog box, enter the desired priority and click **Confirm**. You can set the priority to an integer from 1 to 8. A larger value indicates a higher priority.

  > **Note** If multiple batch synchronization solutions have the same priority, the system runs them based on the order they are committed.

- Delete the batch synchronization solution.

  Find the batch synchronization solution that you want to delete and choose **More** > **Delete** in the Operation column. In the Delete message, click **OK**.

  > **Note** After you click OK, only the configuration record of the batch synchronization solution is deleted. The synchronization nodes generated by the solution and data tables generated by the synchronization nodes are not affected.

4.4.6. Configure and view a real-time sync solution used to synchronize all data in a database

After you configure data sources, network environments, and resource groups, you can create and run a real-time sync solution to synchronize all data in a database. This topic describes how to create a real-time sync solution to synchronize data in specific or all tables in a database to Elasticsearch in offline mode and then synchronize incremental data in the database to Elasticsearch in real time. This topic also describes how to view the statuses of the nodes generated by the real-time sync solution.

Prerequisites

Before you create a data synchronization solution, make sure that the following operations are performed:

- Plan and configure resources
- Configure data sources for data synchronization from MySQL
- Configure data sources for data synchronization from PolarDB
- Add data sources

Context

You can use the hot-warm architecture that is provided by Elasticsearch in a fully managed manner to store the real-time data of enterprises. DataWorks provides real-time sync solutions that can be used to synchronize all data in a database to Elasticsearch in real time based on the architecture. You can use such a solution to first synchronize data in specific or all tables in a database to Elasticsearch in offline mode and then synchronize incremental data in the database to Elasticsearch in real time. You can also view the details of the solution, the statuses of the nodes generated by the solution, and data updates in the database in real time. This facilitates subsequent data searches, analysis, and development.

Real-time sync solutions that are used to synchronize all data in a database provide the following benefits:

- Synchronizes the full data of a database.

  You do not need to create multiple batch data synchronization nodes to synchronize source tables one by one. You can directly create a batch synchronization solution to synchronize some or all of the tables in a database at a time.

- You can configure synchronization rules in a flexible manner.
  - You can configure synchronization rules for different DDL messages based on your business requirements. For example, if you select **Ignore** for a DDL message that is specified in the source and used to drop a table in the destination, the system ignores the message and does not drop the table in the destination when the system receives the message.
  - You can add or remove source tables for a sync solution that is running.
  - You can configure synchronization rules for destination Elasticsearch indexes to determine whether to synchronize the incremental data in source tables to destination Elasticsearch indexes based on your business requirements. After the incremental data is synchronized, the incremental data can be searched in destination Elasticsearch indexes.

- Requires only simple configurations.

  You do not need to perform complex operations, such as creating synchronization nodes, databases, and tables, configuring dependencies for nodes, and configure mappings between sources and destinations. Instead, you need only to configure a batch synchronization solution in a configuration wizard.

- Large amounts of data can be updated in real time. This improves the efficiency of automated O&M.

Scenarios

If you want the system to monitor data updates in business databases in real time, you can use real-time sync solutions to synchronize all data in the databases. This way, upper-layer applications can search for, analyze, and develop data in real time.

Limits

- You can use a real-time sync solution to synchronize all data only from a MySQL or PolarDB database to Elasticsearch.
- A real-time sync solution that is used to synchronize all data in a database can be run only on exclusive resource groups.

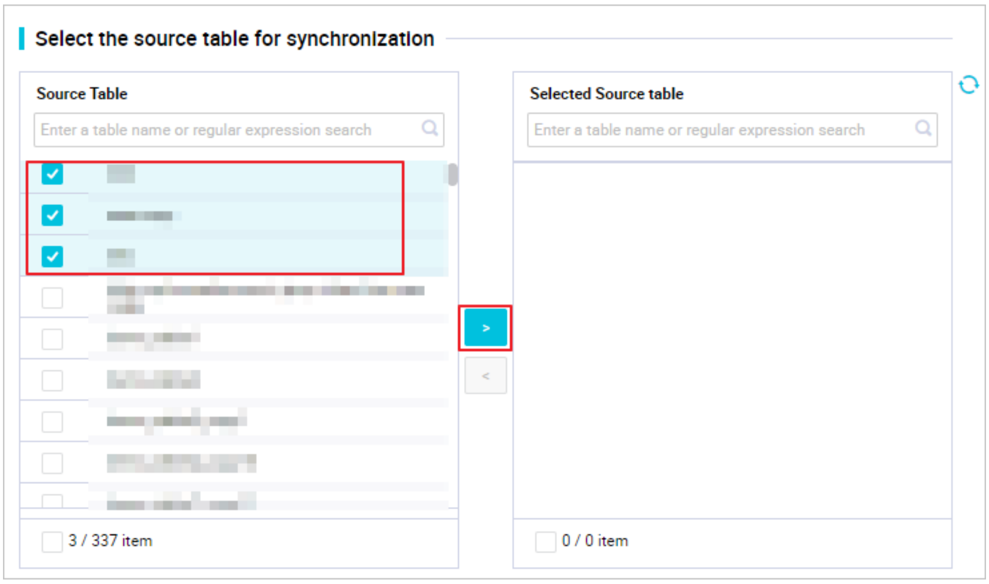
Create a real-time sync solution to synchronize all data in a database

1.
2.
3. In the **Create Data Synchronization Solution** dialog box, click **One-click realtime synchronization to Elasticsearch**.
4.
5. Select a source and configure synchronization rules.
   i. In the **Data Source** section, specify the **Type** and **Data source** parameters.

      ⓘ Note

      You can use a real-time sync solution to synchronize all data only from a MySQL or PolarDB database to Elasticsearch.

ii. In the **Source Table** section, select the tables whose data you want to synchronize from the **Source Table** list. Then, click the > icon to move the tables to the **Selected Source Table** list.



Select the source table for synchronization

Source Table
Enter a table name or regular expression search
3 / 337 item
>
<
Selected Source table
Enter a table name or regular expression search
0 / 0 item

The Source Table list displays all tables in the selected source. You can synchronize data in specific or all tables in the source.

> **Notice** If a selected table has no primary key, you must customize a primary key when you map the table to a destination Elasticsearch index. This primary key is used to remove duplicate data during synchronization. For example, you can use one field or a combination of several fields as the primary key of the table. For more information, see Step 6 in this topic.

iii. In the **Conversion Rule for Table Name** section, click **Add rule** to select a rule.

Supported options include **Conversion Rule for Table Name** and **Rule for Destination Index Name**.

- **Conversion Rule for Table Name**: the rule for converting the names of source tables to those of destination Elasticsearch indexes.
- **Rule for Destination Index Name**: the rule for adding a prefix and a suffix to the converted names of destination Elasticsearch indexes.

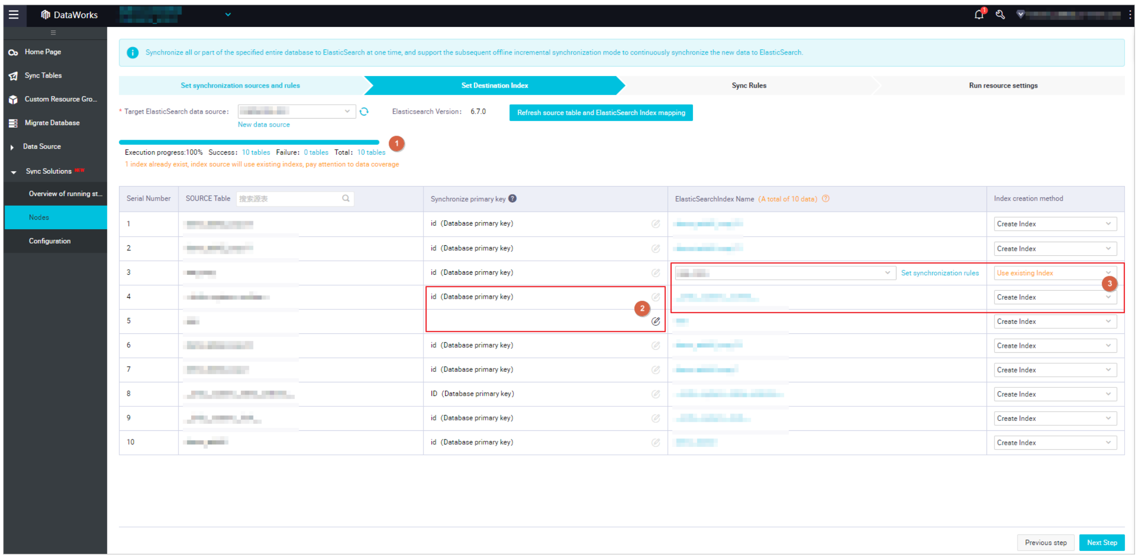
iv. Click **Next Step**.

6. Select the destination and configure destination Elasticsearch indexes.

i. In the **Set Destination Index** step, specify the **Destination** parameter.

ii. Click **Refresh source table and Elasticsearch Index mapping** to configure the mappings between the source tables and destination Elasticsearch indexes.

iii. View the mapping progress, source tables, and mapped destination Elasticsearch indexes.



Synchronize all or part of the specified entire database to ElasticSearch at one time, and support the subsequent offline incremental synchronization mode to continuously synchronize the new data to ElasticSearch.

* Target ElasticSearch data source: [redacted] Elasticsearch Version: 6.7.0

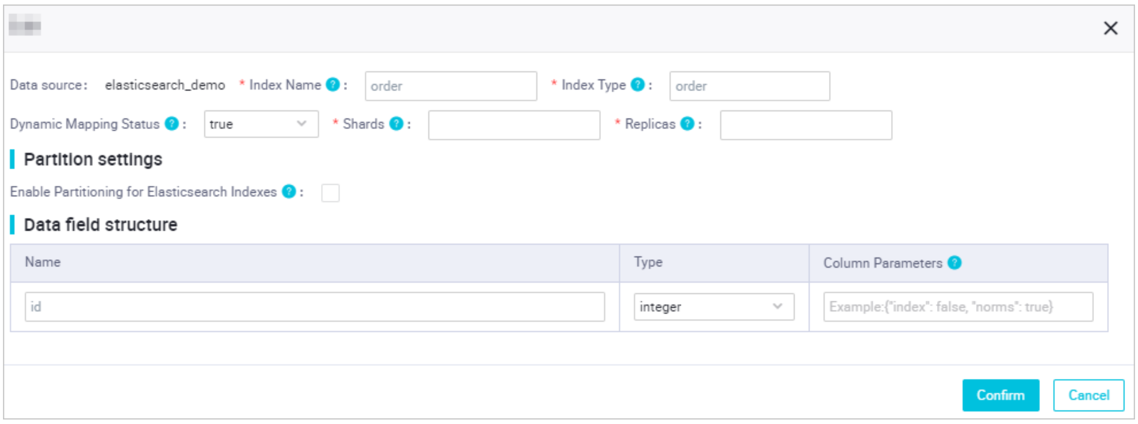
Execution progress:100% Success: 10 tables Failure: 0 tables Total: 10 tables

1 index already exist, index source will use existing indexs, pay attention to data coverage

Serial Number	SOURCE Table 搜索源表	Synchronize primary key	ElasticSearchIndex Name (A total of 10 data)	Index creation method
1	[redacted]	id (Database primary key)	[redacted]	Create Index
2	[redacted]	id (Database primary key)	[redacted]	Create Index
3	[redacted]	id (Database primary key)	[redacted] Set synchronization rules	Use existing Index
4	[redacted]	id (Database primary key)	[redacted]	Create Index
5	[redacted]		[redacted]	Create Index
6	[redacted]	id (Database primary key)	[redacted]	Create Index
7	[redacted]	id (Database primary key)	[redacted]	Create Index
8	[redacted]	ID (Database primary key)	[redacted]	Create Index
9	[redacted]	id (Database primary key)	[redacted]	Create Index
10	[redacted]	id (Database primary key)	[redacted]	Create Index

No.	Description
1	The progress of mapping the source tables to destination Elasticsearch indexes. **Note** The mapping may require a long period of time if you synchronize data from a large number of tables.
2	■ If the tables in the source database contain primary keys, the system removes duplicate data based on the primary keys during the synchronization. ■ If the tables in the source database do not contain primary keys, you can click the ✎ icon to customize primary keys. You can use one field or a combination of several fields as the primary keys of the tables. This way, the system removes duplicate data based on the primary keys during the synchronization.
3	The method that is used to create an index. Valid values: ■ **Create Index**: If you select this method, the name of the Elasticsearch index that is automatically created appears in the **Elasticsearch Index Name** column. You can click the name of the index to change the values of the parameters related to the index. ■ **Use Existing Index**: If you select this method, select the name of the desired index from the drop-down list in the **Elasticsearch Index Name** column. Then, you can click **View Field Mapping** to view the mappings between the source tables and destination Elasticsearch indexes.

If you set the **Index creation method** parameter to **Create Index** , you can click the Elasticsearch index name that appears in the Elasticsearch Index Name column to change the values of the parameters related to the destination Elasticsearch index based on your business requirements.



Data source: elasticsearch_demo * Index Name ?: order * Index Type ?: order

Dynamic Mapping Status ?: true * Shards ?: * Replicas ?:

Partition settings

Enable Partitioning for Elasticsearch Indexes ?: □

Data field structure

Name	Type	Column Parameters ?
id	integer	Example:{"index": false, "norms": true}

- **Dynamic Mapping Status**: specifies whether to dynamically synchronize new fields in the source tables to the destination Elasticsearch indexes during synchronization. Valid values:
  - **true**: If the system detects that the source tables contain new fields, the system synchronizes the fields to the mapped destination Elasticsearch indexes, and the fields can be searched in the indexes after synchronization. Default value: true.
  - **false**: If the system detects that the source tables contain new fields, the system synchronizes the fields to the mapped destination Elasticsearch indexes, but the fields cannot be searched in the indexes after synchronization.
  - **strict**: If the system detects that the source tables contain new fields, the system does not synchronize the fields to the mapped destination Elasticsearch indexes, and an error is reported. You can view the details of the error in the node logs.

  For more information about dynamic mappings, see the description of the dynamic parameter for open source Elasticsearch.
- **Shards** and **Replicas**: the number of primary shards for the destination Elasticsearch index and the number of replica shards for each primary shard. The shards are distributed on different nodes in an Elasticsearch cluster to support distributed searches. This improves the query efficiency of Elasticsearch. For more information, see Terms.

  > ? **Note** The values of the **Shards** and **Replicas** parameters cannot be changed after you specify them and the real-time sync solution starts to run. The default values of the Shards and Replicas parameters are *1*.
- **Partition settings**: You can use a column in a source table as a partition key column. This parameter must be used together with the **Shards** and **Replicas** parameters. By default, the Enable Partitioning for Elasticsearch Indexes check box is not selected.
- **Data field structure**: This section allows you to configure the types and extended attributes of the fields in the mapped destination Elasticsearch indexes. For more information, see Field data types in open source Elasticsearch.

> ? **Note** If you do not change the values of the parameters related to the destination Elasticsearch indexes after the indexes are created, the system synchronizes data based on the default values of the parameters.

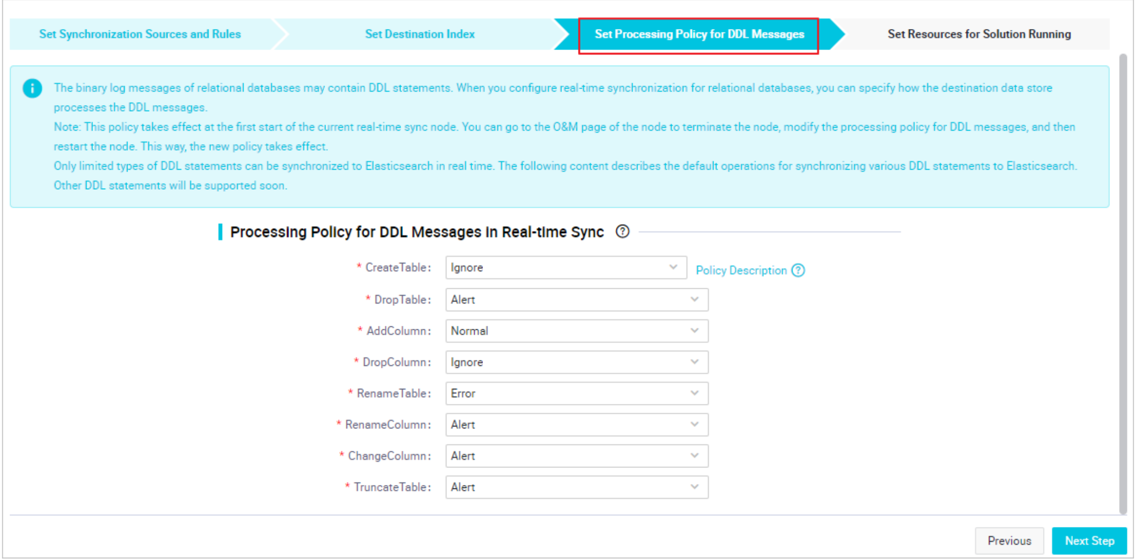
iv. Click **Next Step**.

7. Configure rules to process DDL messages.

   Sources, such as MySQL, PolarDB, may contain multiple DDL messages. Before you synchronize data, you can configure different rules to process DDL messages based on your business requirements.

   > ? **Note** The rules apply when a real-time sync solution is run for the first time. If you want to modify the rules in subsequent operations, go to the configuration page of the real-time sync solution to perform the operation. For more information, see Manage the real-time sync solution.

i. In the **Set Processing Policy for DDL Messages** step, configure rules to process DDL messages during data synchronization.



Set Synchronization Sources and Rules
Set Destination Index
Set Processing Policy for DDL Messages
Set Resources for Solution Running

The binary log messages of relational databases may contain DDL statements. When you configure real-time synchronization for relational databases, you can specify how the destination data store processes the DDL messages.
Note: This policy takes effect at the first start of the current real-time sync node. You can go to the O&M page of the node to terminate the node, modify the processing policy for DDL messages, and then restart the node. This way, the new policy takes effect.
Only limited types of DDL statements can be synchronized to Elasticsearch in real time. The following content describes the default operations for synchronizing various DDL statements to Elasticsearch. Other DDL statements will be supported soon.

Processing Policy for DDL Messages In Real-time Sync

* CreateTable: Ignore
* DropTable: Alert
* AddColumn: Normal
* DropColumn: Ignore
* RenameTable: Error
* RenameColumn: Alert
* ChangeColumn: Alert
* TruncateTable: Alert

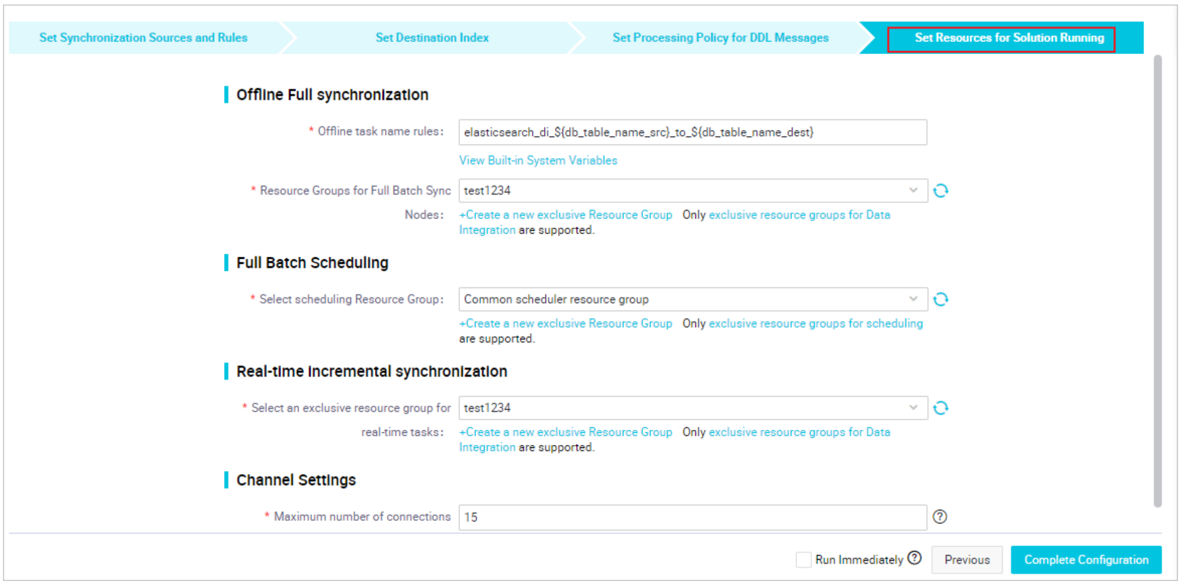
The following table describes the processing rules for different DDL messages.

DDL message	Rule
CreateTable	DataWorks processes a DDL message of the related type based on the following rules after it receives the message: ■ **Normal**: sends the message to the destination. Then, the destination processes the message. Each destination may process DDL messages based on its own business logic. If you select Normal for CreateTable, DataWorks only forwards the messages. ■ **Ignore**: ignores the message and does not send it to the destination. ■ **Alert**: ignores the message and records the alert in real-time synchronization logs. In addition, the alert contains information about the reason indicating that a message is ignored because of a running error. ■ **Error**: returns an error when the real-time sync solution is running and terminates the real-time sync solution.
DropTable	
AddColumn	
DropColumn	
RenameTable	
RenameColumn	
ChangeColumn	
TruncateTable	

ii. Click **Next Step**.

8. Configure the resources required by the sync solution.

In the **Set Resources for Solution Running** step, set the parameters as required.


Set Synchronization Sources and Rules
Set Destination Index
Set Processing Policy for DDL Messages
Set Resources for Solution Running

Offline Full synchronization

* Offline task name rules: elasticsearch_di_${db_table_name_src}_to_${db_table_name_dest}

View Built-in System Variables

* Resource Groups for Full Batch Sync Nodes: test1234

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

Full Batch Scheduling

* Select scheduling Resource Group: Common scheduler resource group

+Create a new exclusive Resource Group Only exclusive resource groups for scheduling are supported.

Real-time incremental synchronization

* Select an exclusive resource group for real-time tasks: test1234

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

Channel Settings

* Maximum number of connections 15

- Offline Sync

Parameter	Description
Offline task name rules	The name of the batch sync node that is used to synchronize the full data of the source. After a sync solution is created, DataWorks first generates a batch sync node to synchronize full data, and then generates real-time sync nodes to synchronize incremental data.
Resource Groups for Full Batch Sync Nodes	The exclusive resource group for Data Integration that is used to run the batch sync node. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Scheduling Settings

Parameter	Description
Select scheduling Resource Group	The resource group for scheduling that is used to run the nodes. Only exclusive resource groups for Data Integration can be used to run sync solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Incremental Sync

Parameter	Description

Parameter	Description
Resource Groups for Incremental Batch Sync Nodes	The exclusive resource group that is used to run the real-time sync nodes. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- **Channel Settings**

Parameter	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: 20.

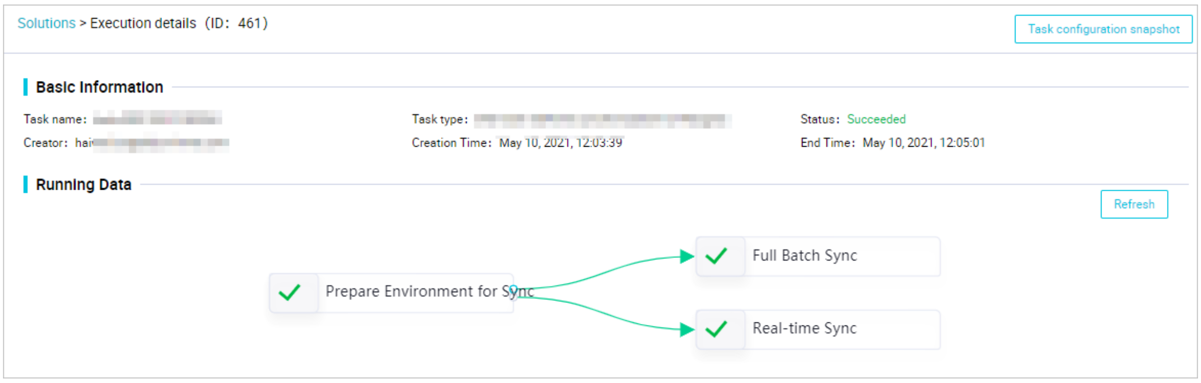
9. Click **Complete Configuration**. The real-time sync solution used to synchronize all data in a database is created.

Run the real-time sync solution

On the **Tasks** page, find the newly created sync solution and choose More > **Submit and Run** in the Operation column to run the sync solution.

View the statuses and results of the sync nodes

- On the **Tasks** page, find the solution that has been run and click **Execution details** in the Operation column. Then, you can view the execution details of all nodes generated by the sync solution.



Solutions > Execution details (ID: 461)

Basic Information

Task name: [redacted]

Task type: [redacted]

Status: Succeeded

Creator: hai[redacted]

Creation Time: May 10, 2021, 12:03:39

End Time: May 10, 2021, 12:05:01

Running Data

- Find a node whose execution details you want to view and click **Execution details** in the Status column. Then, you can click the link provided in the dialog box that appears to go to the DataStudio page.

Manage the real-time sync solution

- View the configurations of the sync solution.

  On the **Tasks** page, find the newly created sync solution and choose **More > View Configuration**. Then, you can view the configurations of the sync solution.

- Modify the sync solution.

  On the **Tasks** page, find the newly created sync solution and choose **More > Modify Configuration**. Then, you can modify the configurations of the sync solution.

  For a sync solution that is successfully run, you can choose **More > Modify Configuration** to add or remove source tables, or set the AddColumn parameter to Normal to synchronize new fields in the source tables to Elasticsearch in real time.

- Add or remove source tables for the sync solution.

  In the **Source Table** section of the **Set Synchronization Sources and Rules** step, add or remove source tables for the sync solution. Then, save the modification and run the sync solution.

- Synchronize new fields in the source tables to Elasticsearch in real time.
  - In the **Set Processing Policy for DDL Messages** step, set the **AddColumn** parameter to **Normal**. Then, the real-time sync solution automatically monitors the column changes in the source tables. If the sync solution detects that the source tables contain new fields, the sync solution automatically synchronizes data in the new fields to the destination Elasticsearch indexes. For more information, see Step 7 in this topic.
  - Set the **Dynamic Mapping Status** parameter to **true**. Then, the sync solution synchronizes new fields in the source tables to the destination Elasticsearch indexes in real time, and the fields can be searched in the future. For more information, see Step 6 in this topic.

- Change the priority for the batch synchronization solution

  Find the newly created batch synchronization solution and choose **More > Change Priority** in the Operation column. In the **Change Priority** dialog box, enter the desired priority and click **Confirm**. You can set the priority to an integer from 1 to 8. A larger value indicates a higher priority.

  > ? **Note** If multiple batch synchronization solutions have the same priority, the system runs them based on the order they are committed.

- Delete the batch synchronization solution.

  Find the batch synchronization solution that you want to delete and choose **More > Delete** in the Operation column. In the Delete message, click **OK**.

  > ? **Note** After you click OK, only the configuration record of the batch synchronization solution is deleted. The synchronization nodes generated by the solution and data tables generated by the synchronization nodes are not affected.

4.5. Synchronize data to Hologres

4.5.1. Plan and configure resources

When you use the sync solutions of DataWorks to synchronize data, you can use only exclusive resource groups for Data Integration to run data integration nodes. In addition, you can select a shared or exclusive resource group for scheduling based on your business requirements. This topic describes the resources that are used for sync solutions and how to configure the resources.

Context

- Resource planning and preparation

  When you use sync solutions to synchronize data, data integration nodes run on resources in resource groups for Data Integration and resource groups for scheduling. In this case, you can use only exclusive resource groups for Data Integration. Before you synchronize data, you must purchase exclusive resources for Data Integration and create an exclusive resource group for Data Integration in your DataWorks workspace.

  For more information about exclusive resource groups for Data Integration, see Exclusive resources for Data Integration.

- Network connections

  An exclusive resource group for Data Integration is essentially a group of Elastic Compute Service (ECS) instances. After you purchase and create such an exclusive resource group, it is isolated from other services. You must associate the resource group with a virtual private cloud (VPC) to ensure network connectivity between the resource group and data sources during subsequent data synchronization.

What's next

After you plan and configure resources, you can configure data sources. You must connect the exclusive resource group for Data Integration to the source and destination. You must also create an account and grant the required permissions to the account. This account is used to access the source and destination. The preceding operations help create sync nodes. You can synchronize data only from PolarDB, Oracle, or MySQL to MaxCompute. You can select a data source based on your business requirements. For more information about how to configure a data source, see Configure a data source (PolarDB), Configure a source Oracle data source, or Configure data sources for data synchronization from MySQL.

4.5.2. Configure a data source (PolarDB)

When you use DataWorks to synchronize data from PolarDB to Hologres, you can refer to the operations described in this topic to configure the network, whitelists, and permissions for data sources to implement data synchronization.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

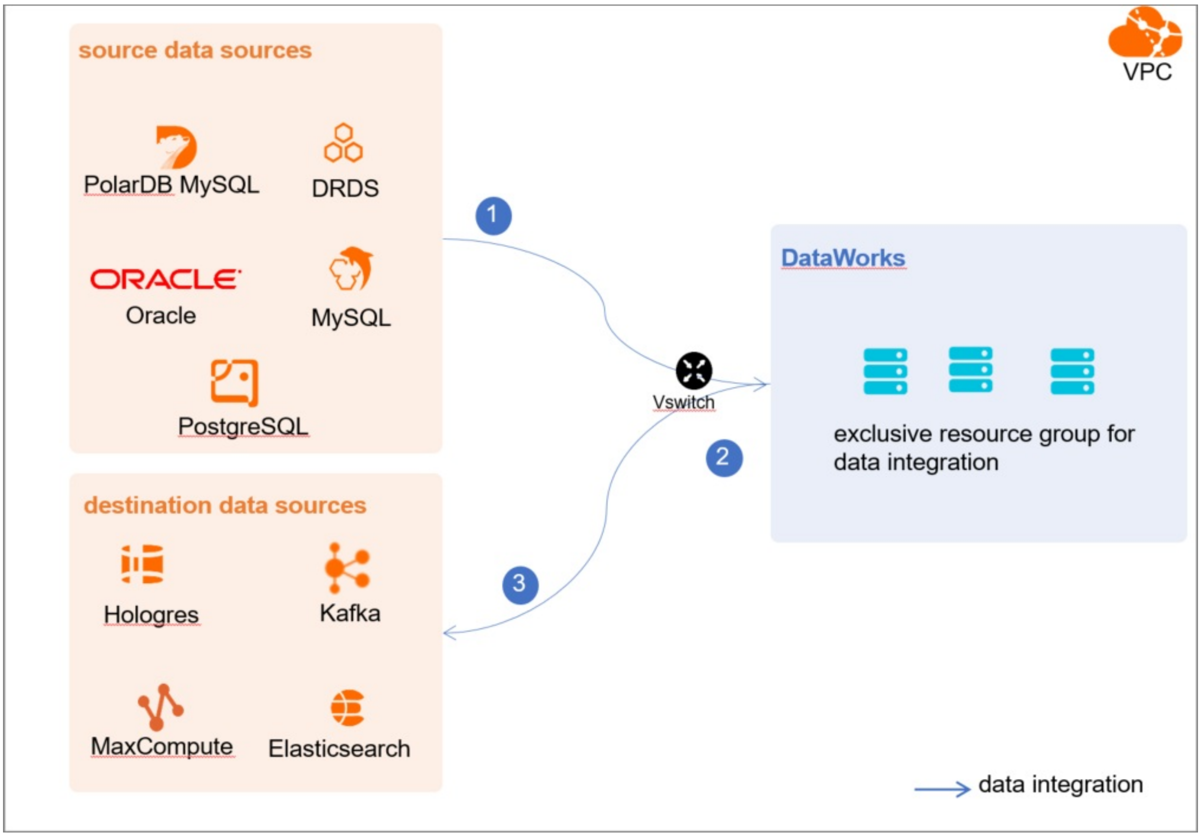
- Prepare data sources: A PolarDB for MySQL cluster and a Hologres instance are created. In this topic, a PolarDB for MySQL cluster is used as the source.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
    - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
    - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source is a PolarDB for MySQL cluster, you must enable the binary logging feature for the cluster. PolarDB for MySQL is fully compatible with MySQL and uses high-level physical logs to replace binary logs. To facilitate the integration between PolarDB and the MySQL ecosystem, you can enable the binary logging feature for PolarDB clusters.

Limits

- Only PolarDB for MySQL clusters can be used as sources in data synchronization solutions. Other types of PolarDB data sources are not supported. In this topic, PolarDB indicates PolarDB for MySQL data sources.
- Only data stored on the primary node of a PolarDB for MySQL cluster can be synchronized.

Procedure

1. Configure a whitelist for the PolarDB for MySQL cluster.

   To add the CIDR block of the VPC where the exclusive resource group for Data Integration resides to a whitelist of the PolarDB for MySQL cluster, perform the following steps:

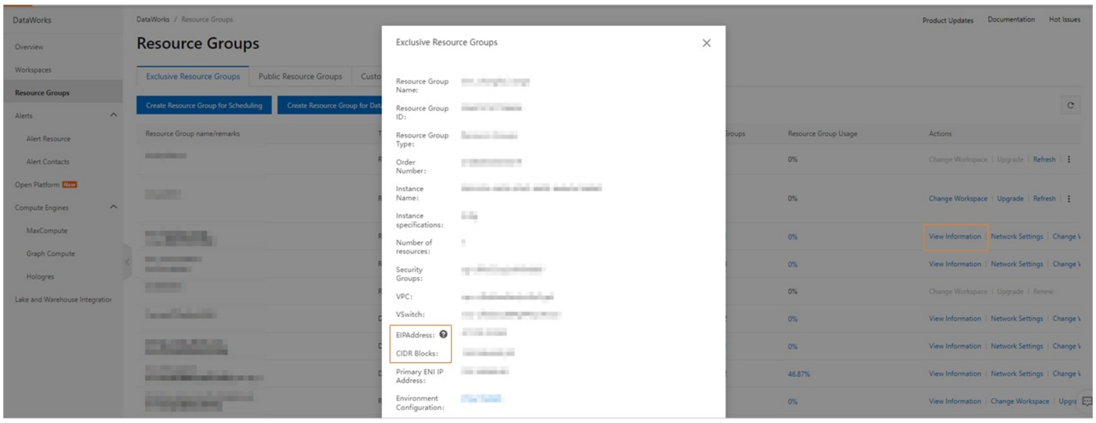
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

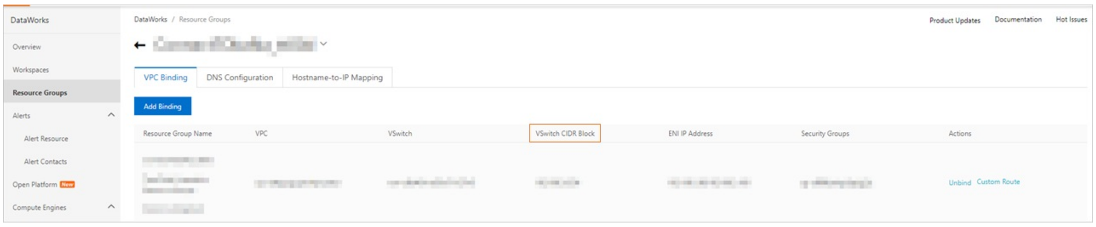
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.

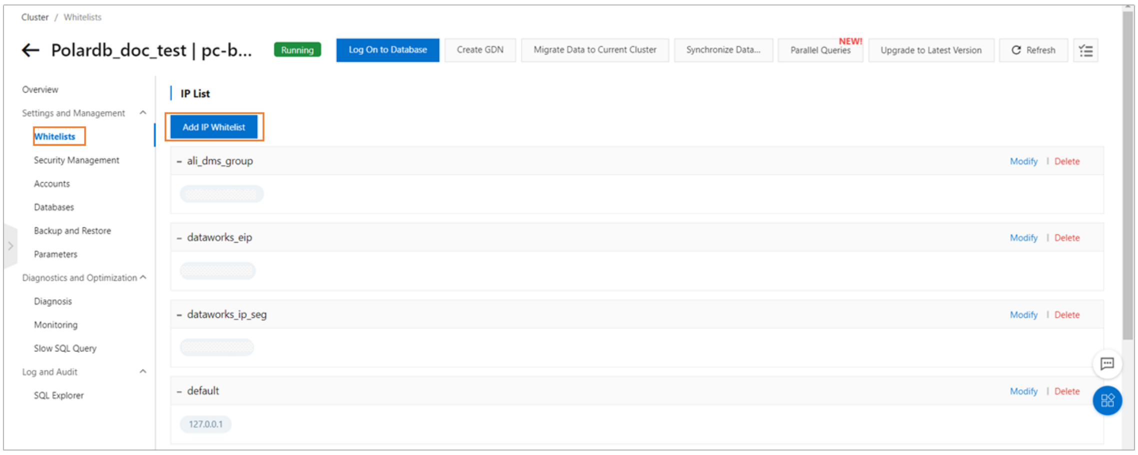


e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR blocks recorded in the preceding steps to the whitelist of the PolarDB for MySQL cluster.



For more information, see Configure an IP whitelist.

2. Create an account and grant the required permissions to the account.

You must create an account to log on to the database of the PolarDB for MySQL cluster. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

i. Create an account.

For more information, see Create a database account.

ii. Grant the required permissions to the account.

You can run the following command to grant the required permissions to the account, or you can directly assign the `SUPER` role to the account.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Account for data synchronization';
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%';
```

3. Enable the binary logging feature for the PolarDB for MySQL cluster.

   For more information, see Enable binary logging.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

4.5.3. Configure a source Oracle data source

When you synchronize data in an Oracle database to Hologres, you can refer to the operations in this topic to prepare configurations such as network environments and whitelists for data sources.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

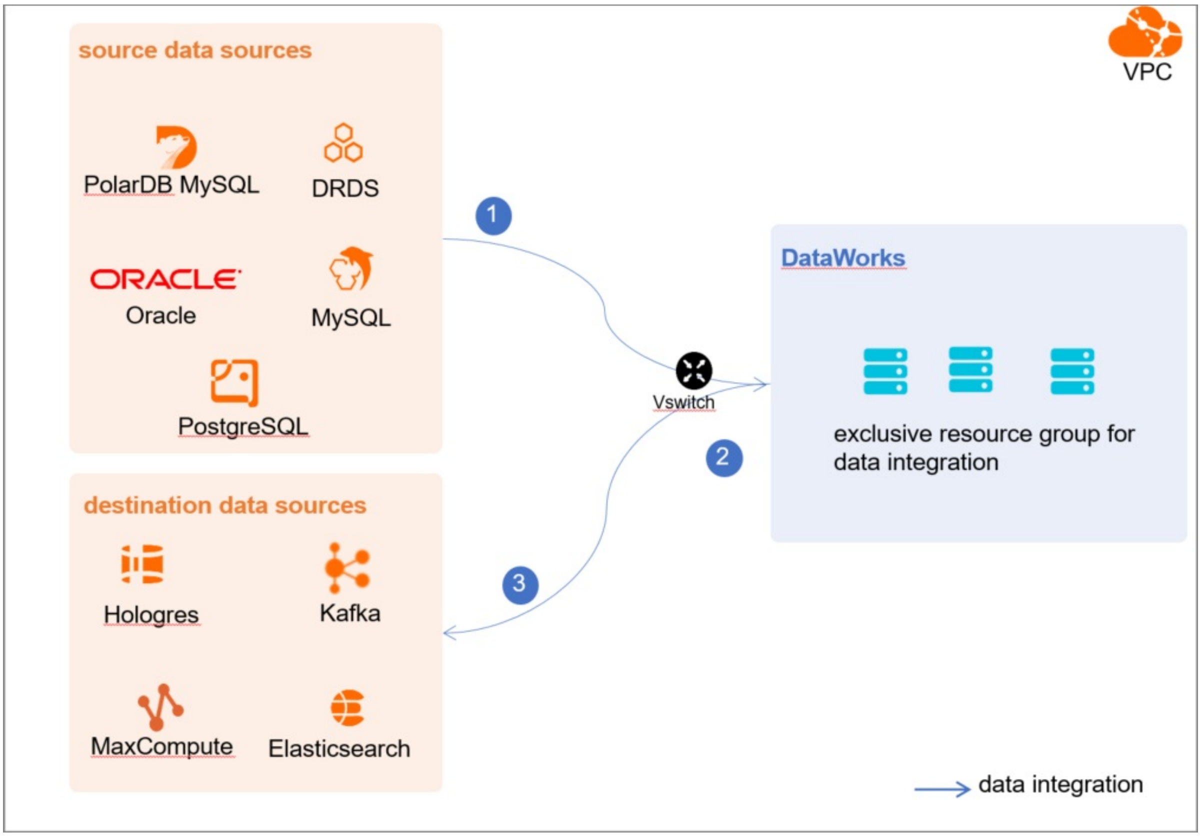
- Prepare data sources: A source Oracle data source and a destination Hologres data source are created.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from source data sources to destination data sources, make sure that data sources and exclusive resource groups for data integration are connected. You must also make sure that the exclusive resource groups for data integration can be used to access the data sources. In addition, the Oracle data source must contain only the character encoding formats and data types that are supported for data integration.

- Configure whitelists for the data sources

If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable generation of database-level redo log files and enable database-level supplemental logging

  You must enable generation of database-level redo log files and enable database-level supplemental logging for the Oracle database to be configured as a source data source.

  - Redo log files: Oracle uses redo log files to ensure that database transactions can be re-executed. This way, data can be recovered in the case of a failure such as power outage.
  - Supplemental logging: Supplemental logging is used to supplement the information recorded in redo log files. In Oracle, a redo log file is used to record the values of the fields that are modified. Supplemental logging is used to supplement the change history in the redo log file. This ensures that the redo log file contains complete information that describes data changes. If operations such as data recovery and data synchronization are performed, you can view complete statements and data updates. Some features of the Oracle database can be better implemented after supplemental logging is enabled. Therefore, you must enable supplemental logging for the database.

    For example, if you do not enable supplemental logging, after you execute the UPDATE statement, the redo log file records only the values of the fields that are modified when the UPDATE statement is executed. If you enable supplemental logging, the redo log file records the values of fields before and after a modification. The conditions that are used to modify destination fields are also recorded. When an exception such as power outage occurs in the database, you can recover data based on the modification details.

    We recommend that you enable supplemental logging for primary key columns or unique index columns.

    - After you enable supplemental logging for primary key columns, the columns that compose a primary key are recorded in logs if the database is updated.
    - After you enable supplemental logging for unique index columns, the columns that compose a unique key or bitmap index are recorded in logs if a column is modified.

- Check character encoding formats

You must make sure that the Oracle database contains only the character encoding formats that are supported for data integration to prevent a data synchronization failure. The following encoding formats are supported for data synchronization: UTF-8, AL32UTF8, AL16UTF16, and ZHS16GBK.

- Check data types

  You must make sure that the Oracle database contains only the data types that are supported for data integration to prevent data synchronization failures. The following data types are not supported for real-time data synchronization: LONG, BFILE, LONG RAW, and NCLOB.

Limits

- You can configure the supplemental logging feature only in a primary Oracle database. Supplemental logging can be enabled for a primary or secondary database.
- The following encoding formats are supported for data synchronization: UTF-8, AL32UTF8, AL16UTF16, and ZHS16GBK.
- The following data types are not supported for real-time data synchronization: LONG, BFILE, LONG RAW, and NCLOB.

Procedure

1. Configure a whitelist for an Oracle database.

   Add the CIDR block of the VPC where the exclusive resource group resides to the whitelist of the Oracle database.

   i. View and record the EIP and CIDR block of the exclusive resource group.

   ii. Add the EIP and CIDR block of the exclusive resource group to the whitelist of the Oracle database.

2. Create an account and grant the required permissions to the account.

   You must create an account to log on to the database. The account must have the required permissions on the Oracle database.

   i. Create an account.

   For more information, see Create an Oracle database account.

   ii. Grant permissions to the account.

   You can run the following commands to grant permissions to the account. Replace `Account for data synchronization` with the created account when you execute the specific statement.

```
grant create session to 'Account for data synchronization';  // Authorize the synchronization account to access the database.
grant connect to 'Account for data synchronization';  // Authorize the synchronization account to connect to the database.
grant select on nls_database_parameters to 'Account for data synchronization';  // Authorize the synchronization account to query the settings of nls_database_parameters.
grant select on all_users to 'Account for data synchronization';  // Authorize the synchronization account to query all users in the database.
grant select on all_objects to 'Account for data synchronization';  // Authorize the synchronization account to query all objects in the database.
grant select on DBA_MVIEWS to 'Account for data synchronization';  // Authorize the synchronization account to check the materialized view of the database.
grant select on DBA_MVIEW_LOGS to 'Account for data synchronization';  // Authorize the synchronization account to view the materialized view logs of the database.
grant select on DBA_CONSTRAINTS to 'Account for data synchronization';  // Authorize the synchronization account to view the constraints on all tables of the database.
grant select on DBA_CONS_COLUMNS to 'Account for data synchronization';  // Authorize the synchronization account to view information about all columns under specified constraints on all tables of the database.
grant select on all_tab_cols to 'Account for data synchronization';  // Authorize the synchronization account to view information about columns in tables, views, and clusters of the database.
grant select on sys.obj$ to 'Account for data synchronization';  // Authorize the synchronization account to view objects in the database. sys.obj$ indicates an object table that is contained in the data dictionary table. The object table contains all objects.
grant select on SYS.COL$ to 'Account for data synchronization';  // Authorize the synchronization account to view definitions of columns in tables of the database. SYS.COL$ stores column definitions.
grant select on sys.USER$ to 'Account for data synchronization';  // Authorize the synchronization account to view the system table of the database. sys.USER$ indicates a default user session service.
grant select on sys.cdef$ to 'Account for data synchronization';  // Authorize the synchronization account to view the system table of the database.
grant select on sys.con$ to 'Account for data synchronization';  // Authorize the synchronization account to view the constraints of the database. sys.con$ records the constraints.
grant select on all_indexes to 'Account for data synchronization';  // Authorize the synchronization account to view all indexes of the database.
grant select on v_$database to 'Account for data synchronization';  // Authorize the synchronization account to check the v_$database view of the database.
grant select on V_$ARCHIVE_DEST to 'Account for data synchronization';  // Authorize the synchronization account to check the V_$ARCHIVE_DEST view of the database.
grant select on v_$log to 'Account for data synchronization';  // Authorize the synchronization account to check the v_$log view of the database. v_$log displays log information about control files.
grant select on v_$logfile to 'Account for data synchronization';  // Authorize the synchronization account to check the v_$logfile view of the database. v_$logfile contains information about redo log files.
grant select on v_$archived_log to 'Account for data synchronization';  // Authorize the synchronization account to check the v$archived_log view of the database. v$archived_log contains information about archived logs.
grant select on V_$LOGMNR_CONTENTS to 'Account for data synchronization';  // Authorize the synchronization account to check the V_$LOGMNR_CONTENTS view of the database.
grant select on DUAL to 'Account for data synchronization';   // Authorize the synchronization account to view the DUAL table of the database. DUAL is a virtual table that contains SELECT syntax rules. In Oracle, only one record is retained in the DUAL table.
grant select on v_$parameter to 'Account for data synchronization';  // Authorize the synchronization account to check the v_$parameter view of the database. v$parameter is a dynamic dictionary table that stores the values of parameters in the database.
grant select any transaction to 'Account for data synchronization';  // Authorize the synchronization account to view transactions of the database.
grant execute on SYS.DBMS_LOGMNR to 'Account for data synchronization';  // Authorize the synchronization account to use the LOGMNR tool. The LOGMNR tool helps you analyze transactions and retrieve lost data.
grant alter session to 'Account for data synchronization';  // Authorize the synchronization account to modify connection configurations of the database.
grant select on dba_objects to 'Account for data synchronization';  // Authorize the synchronization account to view all objects of the database.
grant select on v_$standby_log to 'Account for data synchronization';  // Authorize the synchronization account to check the v_$standby_log view of the database. v_$standby_log contains archived logs of the secondary database.
grant select on v_$ARCHIVE_GAP to 'Account for data synchronization';  // Authorize the synchronization account to query missing archived logs.
```

To synchronize full offline data, you must also run the following command to grant the query permission on all tables to the synchronization account:

```
grant select any table to 'Account for data synchronization';
```

In Oracle 12c and later, you must run the following command to grant the log mining permission to the synchronization account. The log mining feature is built in Oracle versions earlier than 12c. You do not need to run the command in these versions.

```
grant LOGMINING TO 'Account for data synchronization';
```

3. Enable supplemental logging and switch a redo log file.

   Log on to the primary database and perform the following steps:

   i. Enable supplemental logging.

      SQL statements:

```
alter database add supplemental log data(primary key) columns; // Enable supplemental logging for primary key columns.
alter database add supplemental log data(unique) columns; // Enable supplemental logging for unique index columns.
```

ii. Switch a redo log file.

After you enable supplemental logging, you must run the following command multiple times to switch a redo log file. We recommend that you run the following command for five times:

```
alter system switch logfile;
```

> **Note** This ensures that data can be written to the next log file after the current log file is full. Data about historical operations will not be lost. This facilitates data recovery.

4. Check character encoding formats of the database.

   Run the following command to check character encoding formats of the database:

   ```
   select * from v$nls_parameters where PARAMETER IN ('NLS_CHARACTERSET', 'NLS_NCHAR_CHARACTERSET');
   ```

   - v$nls_parameters stores values of parameters in the database.
   - NLS_CHARACTERSET indicates a database character set. NLS_NCHAR_CHARACTERSET indicates a national character set. These two sets are used to store data of the character type.

   The following encoding formats are supported for data synchronization: UTF-8, AL32UTF8, AL16UTF16, and ZHS16GBK. If the database contains the character encoding formats that are not supported for data synchronization, change the formats before you synchronize data.

5. Check the data types of tables in the database.

   You can execute the SELECT statement to query the data types of tables in the database. Sample statement that is executed to query the data types of the *'tablename'* table:

   ```
   select COLUMN_NAME,DATA_TYPE from all_tab_columns where TABLE_NAME='tablename';
   ```

   - COLUMN_NAME: the name of the column.
   - DATA_TYPE: the data type of the column.
   - all_tab_columns: the view that stores information about all columns in tables of the database.
   - TABLE_NAME: the name of the destination table for a query. When you execute the preceding statement, replace *'tablename'* with the name of the destination table for a query.

   You can also execute the `select * from 'tablename';` statement to query the information about the destination table and obtain data types.

   The following data types are not supported for real-time data synchronization: LONG, BFILE, LONG RAW, and NCLOB. If a table contains one of these data types, remove the table from the real-time synchronization solution list or change the data type before you synchronize data.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

4.5.4. Configure data sources for data synchronization from MySQL

When you synchronize data in a MySQL database to Hologres, you can refer to the operations in this topic to prepare configurations such as network environments and whitelists for data sources.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

- Prepare data sources: A source MySQL data source and a destination Hologres data source are created.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.

- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.
- Execute the following statement to check the version of your MySQL database. You can use a `MySQL 5.x` or `MySQL 8.x` database.

```
select version();
```

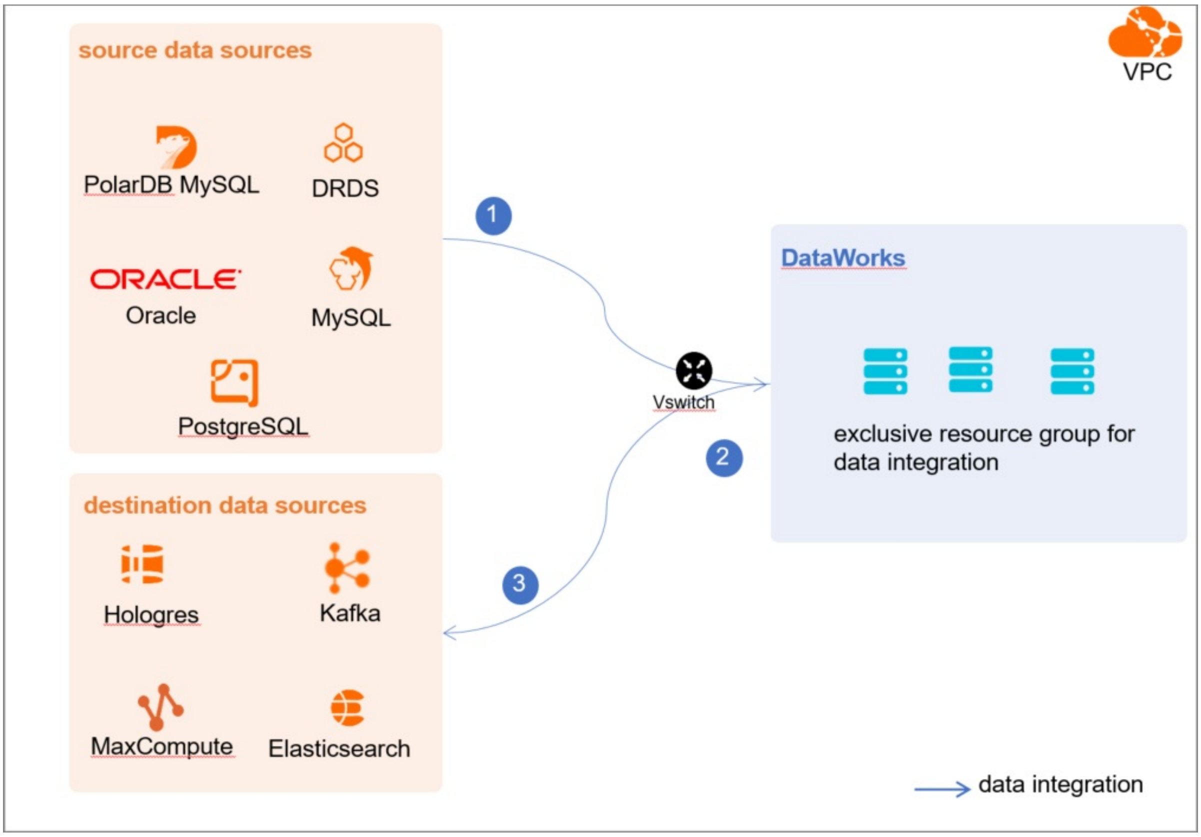
> Note Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.x` or `V8.x`. PolarDB for MySQL is not supported. If the MySQL version of your ApsaraDB RDS for MySQL database is not `V5.x` or `V8.x`, use an ApsaraDB RDS for MySQL database whose MySQL version is `V5.x` or `V8.x`. Otherwise, the data synchronization node fails to run.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

  If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source data source is a MySQL database, you must enable the binary logging feature. Binary logs record changes to all table schemas and modifications on table data. You can execute statements such as CREATE and ALTER to perform operations on table schemas. You can execute statements such as INSERT, UPDATE, and DELETE to perform operations on table data. You can use binary logs to view the change history of the database, back up incremental data and restore data in the database, and replicate data from the primary database to secondary databases.

  Formats of binary logs:

  - Statement: SQL statement-based replication. Binary logs in this format record the SQL statements that are executed to modify data entries.
  - Row: row-based replication. Binary logs in this format record only modification details about data entries in rows.
  - Mixed: replication in mixed mode. This mode combines the statement and row formats. In most cases, binary logs in the statement format are used to record the SQL statements that are executed to modify data entries, such as functions. If data replication from the primary database to secondary databases cannot be implemented by using binary logs in this format, switch to the row format. MySQL determines which format to use based on each SQL statement that is executed.

Limits

Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.X` or `V8.X`. PolarDB for MySQL is not supported.

Procedure

1. Configure a whitelist for the MySQL database.

   Add the CIDR block of the VPC in which the exclusive resource group resides to the whitelist of the MySQL database.

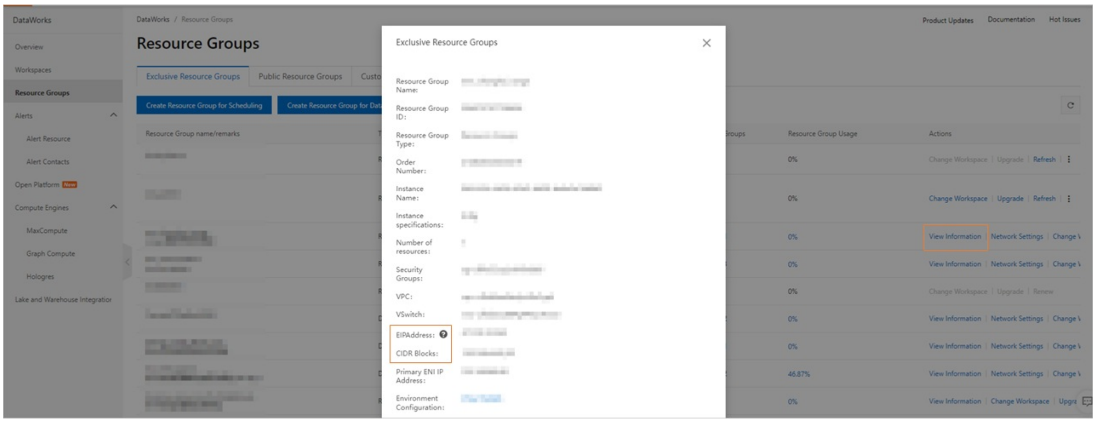
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

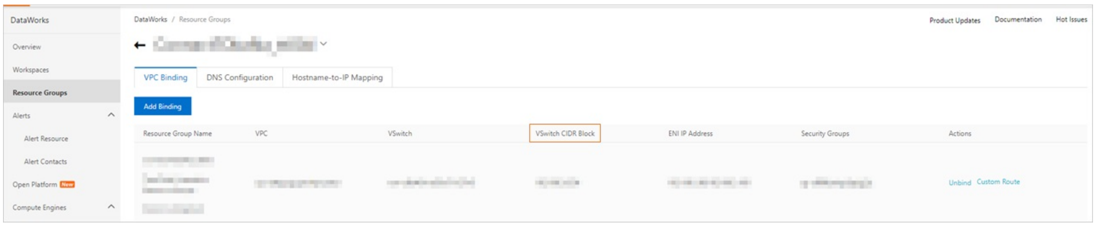
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR block of the exclusive resource group to the whitelist of the MySQL database.

2. Create an account and grant permissions to the account.

You must create an account to log on to the MySQL database. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

i. Create an account.

For more information, see Create an account to access a MySQL database.

ii. Grant permissions to the account.

You can run the following command to grant permissions to the account. Alternatively, you can grant the `SUPER` permission to the account. Replace `Account for data synchronization` with the created account when you execute the specific statement.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Password'; // Create an account that is used to synchronize data and set a password so that you can use the account and password to access the database from a host. % indicates a host.
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%'; // Grant the SELECT, REPLICATION SLAVE, and REPLICATION CLIENT permissions to the account.
```

`*.*` indicates that the synchronization account is granted the preceding permissions on all tables in all databases. You can also grant the preceding permissions on specified tables in the specified database to the synchronization account. For example, to grant the account the preceding permissions on the *user* table in the *test* database, execute the following statement: `GRANT SELECT, REPLICATION CLIENT ON test.user TO 'Account for data synchronization'@'%';` .

> **Note** The `REPLICATION SLAVE` permission is a global permission. You cannot grant this permission on specified tables in the specified database to the synchronization account.

3. Enable the binary logging feature for the MySQL database.

   Perform the following steps to check whether the binary logging feature is enabled and to query the format of binary logs:

   - Execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_bin";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled.

   - If you use a secondary database to synchronize data, execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_slave_updates";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled for the secondary database.

   If the returned result is different from the preceding result, follow the instructions described in the *MySQL documentation* to enable the binary logging feature.

   Execute the following statement to view the format of binary logs:

   ```
   show variables like "binlog_format";
   ```

   Returned result:

   - *ROW*: The format of binary logs is row.
   - *STATEMENT*: The format of binary logs is statement.
   - *MIXED*: The format of binary logs is mixed.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

4.5.5. Configure data sources for data synchronization from PolarDB-X

When you synchronize data from a PolarDB-X database to Hologres, you can refer to the operations in this topic to prepare configurations such as network environments, whitelists, and permissions for data sources.

Prerequisites

Before you configure the data sources, make sure that the following operations are performed:

- Prepare data sources: A source PolarDB-X data source and a destination Hologres data source are created.

  > **Note** The PolarDB-X data source must meet the following requirements:
  >
  > - The instance must be a non-read-only instance of PolarDB-X V1.0.
  > - The data source can be connected by using the Alibaba Cloud instance mode. Otherwise, the sync node that is used to synchronize data from the PolarDB-X instance fails.
  > - The storage type must be PolarDB for MySQL and ApsaraDB RDS (excluding ApsaraDB RDS for MySQL). ApsaraDB RDS can be used only for existing PolarDB-X instances and cannot be used for newly purchased PolarDB-X instances.
  >
  > For more information about how to create a PolarDB-X V1.0 instance, see *Create a PolarDB-X 1.0 instance* or Create a instance in PolarDB-X documentation.

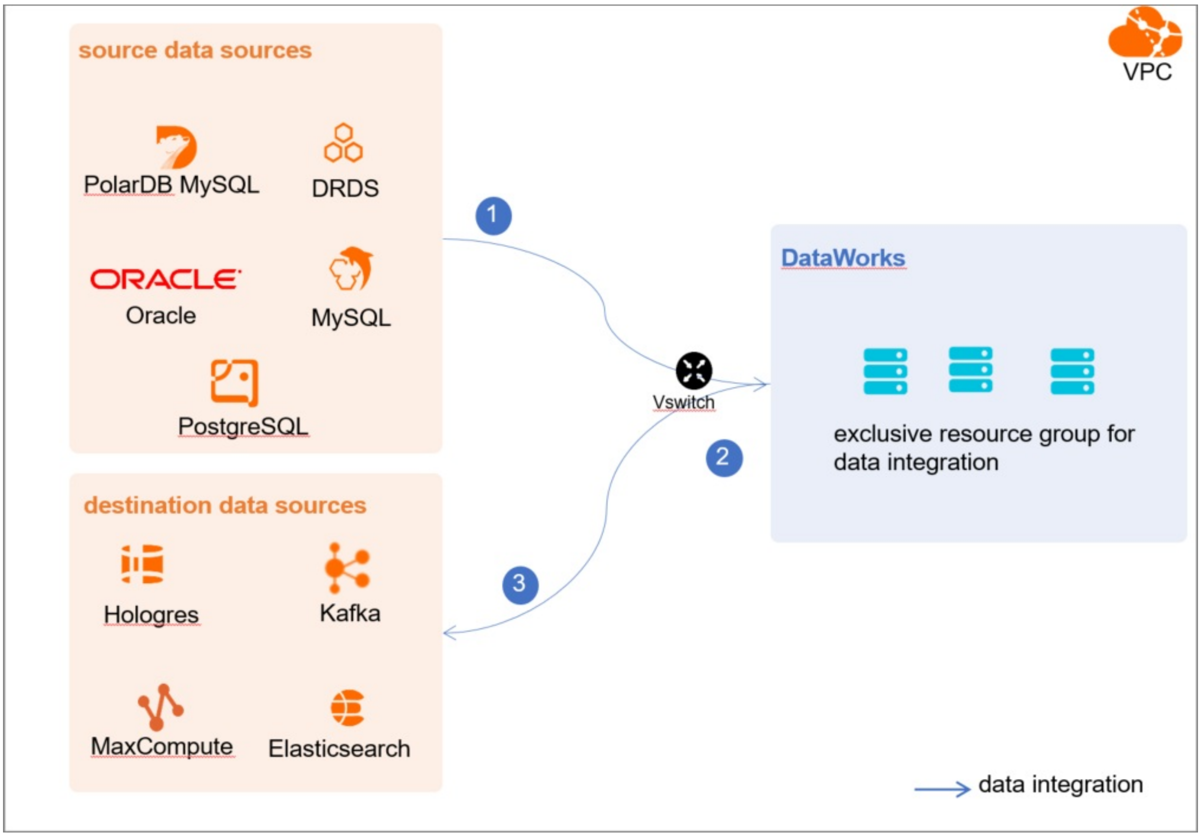
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and exclusive resource groups for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

Procedure

1. Configure a whitelist for the PolarDB-X instance.

   Add the CIDR block of the VPC where the exclusive resource group resides to a whitelist of the PolarDB-X instance. You can perform the following operations:

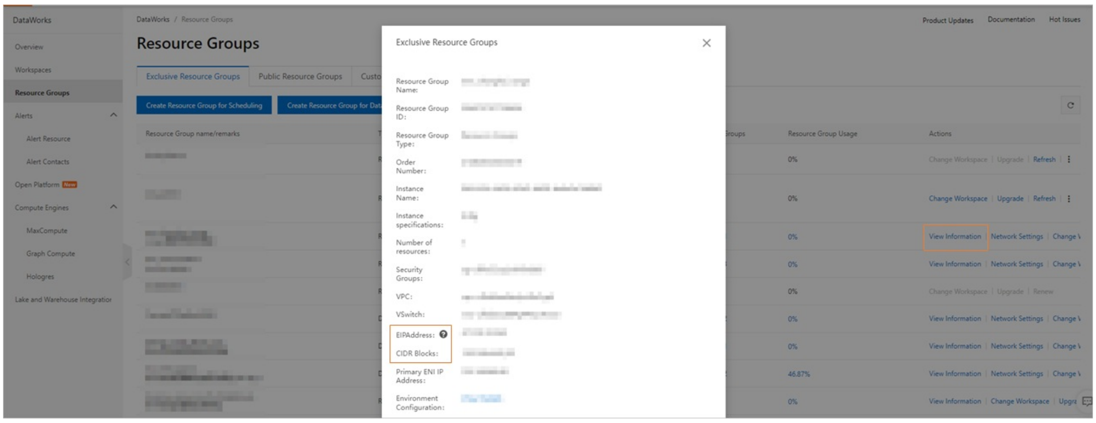
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

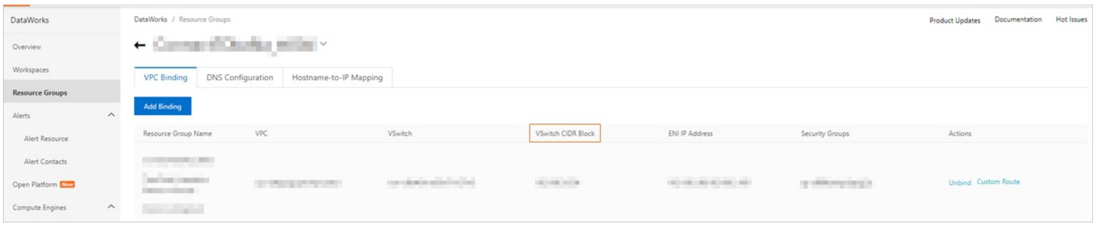
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration to a whitelist of the PloarDB-X instance.

For more information, see Set an IP address whitelist.

2. Create an account and grant the required permissions to the account.

You need to create an account to log on to the databases of the PolarDB-X instance for subsequent operations. For more information, see Manage accounts.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

4.5.6. Configure a source PostgreSQL data source

When you use DataWorks to synchronize data from a PostgreSQL database to Hologres, you can refer to the operations in this topic to configure the network, whitelists, and permissions for data sources to implement data synchronization.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

- Data sources are purchased. A source PostgreSQL data source and a destination Hologres data source are purchased.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more

information, see Plan and configure resources.

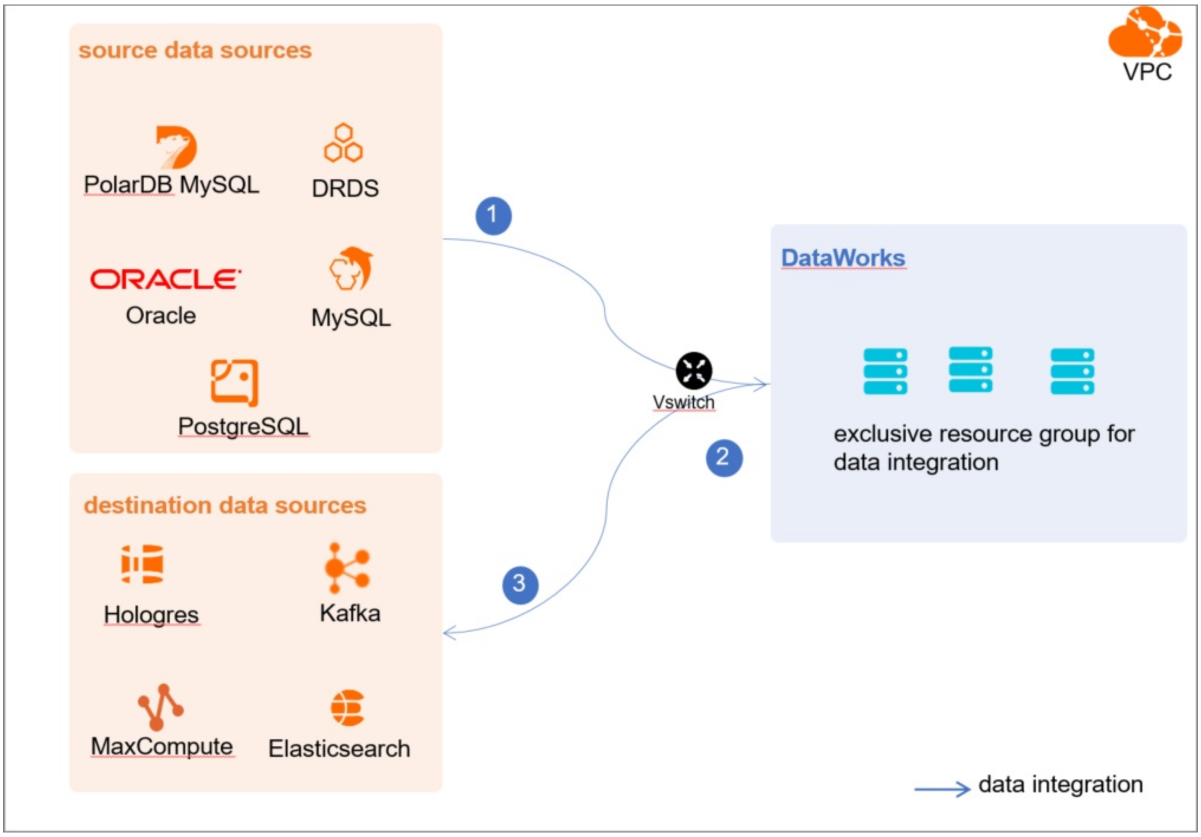
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and exclusive resource groups for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

  If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Check whether the database version is the version supported by real-time synchronization nodes of Data Integration in DataWorks.

  The following PostgreSQL database versions are supported: PostgreSQL `10` , `11` , `12` and `14.1` . You can execute the following statement to view the version of the PostgreSQL database:

  ```
  show server_version
  ```

Limits

Real-time synchronization nodes of Data Integration have the following limits:

- Data Integration supports `ADD COLUMN` statements.
    - An `ADD COLUMN` statement cannot be combined with other DDL statements, such as `DROP COLUMN` .

    > **Notice** If you execute an `ADD COLUMN` statement together with an `ALTER COLUMN` statement, such as a `DROP COLUMN or RNAME COLUMN` statement on a data synchronization node, the node cannot normally work.

    - DDL statements except `ADD COLUMN` statements cannot be identified.
- `ALTER TABLE and CREATE TABLE` statements are not supported.
- TEMPRORARY tables and UNLOGGED tables cannot be replicated. The PostgreSQL database does not provide a mechanism for log parsing subscriptions to these two types of tables.
- Sequences cannot be replicated (`serial, bigserial, and identity`).
- TRUNCATE statements are not supported.
- Large objects cannot be replicated (BYTEA).
- Views, materialized views, and foreign tables cannot be replicated.

Procedure

1. Configure a whitelist for the PostgreSQL database.

   Add the CIDR block of the virtual private cloud (VPC) where the exclusive resource group resides to the whitelist of the PostgreSQL database.

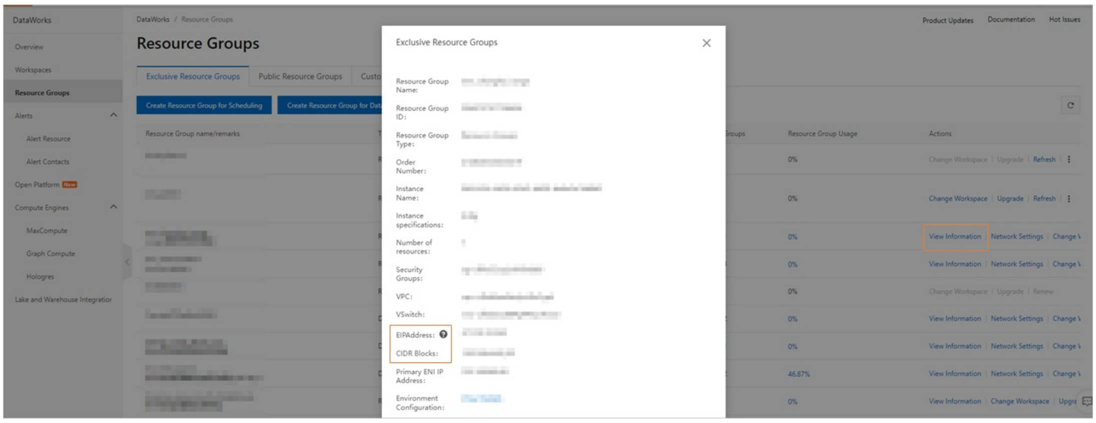
   i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

      a. Log on to the DataWorks console.

      b. In the left-side navigation pane, click **Resource Groups**.

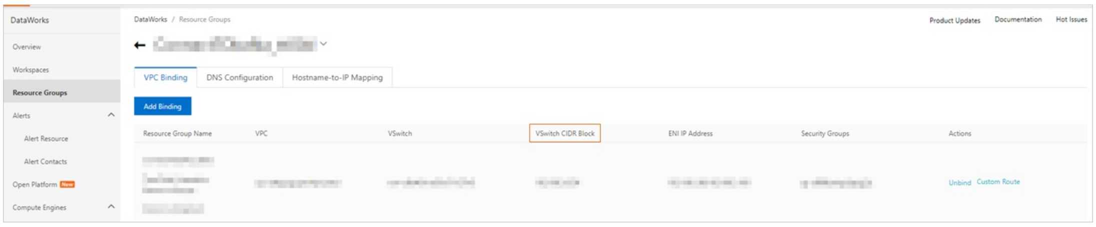
      c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

      d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



      e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

      f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



    ii. Add the elastic IP address (EIP) and CIDR block of the exclusive resource group to the whitelist of the PostgreSQL database.

2. Create an account and grant the required permissions to the account.

   You must create an account to log on to the PostgreSQL database for subsequent operations. The account must have the `REPLICATION and LOGIN` permissions on the PostgreSQL database.

   > Note
   >
   > Real-time synchronization supports only the logical replication mechanism. Logical replication uses a publish and subscribe model in which one or more subscribers subscribe to one or more publications on the node of the publisher. The subscribers extract data from the publications to which they subscribe.
   >
   > Logical replication of a table usually starts with creating a snapshot for the data in the database of the publisher and replicating the snapshot to the subscriber. After logical replication is complete, modifications on the publisher are sent to the subscriber in real time.

    i. Create an account.

    For more information, see Create a database and an account on an ApsaraDB RDS for PostgreSQL instance.

    ii. Grant the required permissions to the account.

    Execute the following statement to check whether the account has the `REPLICATION` permission on the PostgreSQL database:

    ```
    select userepl from pg_user where usename='xxx'
    ```

    The expected return result is True. If False is returned, the account does not have the REPLICATION permission on the PostgreSQL database. You can execute the following statement to grant the REPLICATION permission to the account:

    ```
    ALTER USER <user> REPLICATION;
    ```

3. Execute the following statement to check whether real-time synchronization supports secondary databases:

   ```
   SELECT pg_is_in_recovery()
   ```

   Real-time synchronization supports only primary databases. The expected return result is False. If True is returned, it indicates that the data source is a secondary database. Real-time synchronization does not support secondary databases. You must modify the configuration of the data source to the primary database. For more information, see Add a PostgreSQL data source.

4. Check whether the value of the `wal_level` parameter is `logical` .

   ```
   show wal_level
   ```

   The `wal_level` parameter specifies the `wal_log` level. The expected return result is logical. If logical is not returned, the logical replication mechanism is not supported.

5. Check whether the `wal_sender` process can be started.

   ```
   -- Query the value of the max_wal_senders parameter.
   show max_wal_senders;
   -- Query the number of pg_stat_replication.
   select count(*) from pg_stat_replication
   ```

   If the `max_wal_senders` parameter is not empty and the value of the `max_wal_senders` parameter is greater than the number of `pg_stat_replication` , the `wal_sender` process can be used. The PostgreSQL database starts the `wal_sender` process for the data synchronization program to send logs to subscribers.

4.5.7. Add a data source

Before you configure and run a data sync node to synchronize data from a source data source to a destination Hologres data source, you must add the source data source and the destination data source in the DataWorks console.

Prerequisites

Before you add data sources, make sure that the following operations are performed:

- Prepare data sources: A source data source and a destination data source are created.

- Create and grant permission to an account: An account that is used to access data sources is created.

Precautions

DataWorks provides workspaces in basic mode and standard mode. A workspace in basic mode does not isolate the development environment from the production environment. A workspace in standard mode isolates the development environment from the production environment.

If you use a workspace in standard mode, you must separately add data sources to the development environment and production environment.

Add a source PolarDB for MySQL data source

To add a source PolarDB for MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source PolarDB for MySQL data source, see Add a PolarDB data source.

If the source PolarDB for MySQL data source that you want to add fails the connectivity test, follow the instructions described in What do I do if the PolarDB data source cannot be connected? to handle the exception.

Add a source Oracle data source

To add a source Oracle data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source Oracle data source, see Add an Oracle data source.

Add a source MySQL data source

To add a source MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source MySQL data source, see Add a MySQL data source.

Add a source PolarDB-X data source

To add a source PolarDB-X data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source PolarDB-X data source, see Add a DRDS data source.

Add a destination Hologres data source

For more information about how to add a destination Hologres data source, see Add a Hologres data source.

What's next

After you add data sources, you can create and run a data sync node to synchronize data from the source data source to the destination data source.

For more information, see Create and configure a sync solution.

4.5.8. Create and configure a sync solution

After you configure data sources, network environments, and resource groups, you can create and run a sync solution. This topic describes how to create a sync solution and view the status of the nodes that are generated by the sync solution.

Prerequisites

Before you create a sync solution, make sure that the following operations are performed:

- Plan and configure resources
- Configure a data source (PolarDB)
- Configure a source Oracle data source
- Configure data sources for data synchronization from MySQL
- Configure data sources for data synchronization from PolarDB-X
- Add a data source

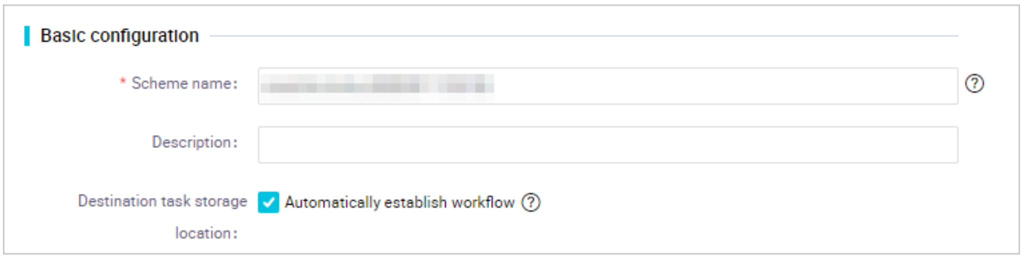
Configure a sync solution

1. Go to the Data Integration page and choose **Sync Solutions > Tasks** to go to the Tasks page.

   For more information, see Select a synchronization solution.

2. On the **Tasks** page, click **New task** in the upper-right corner.
3. In the **Create Data Synchronization Solution** dialog box, click **One-click real-time synchronization to Hologres**.
4. In the Set Synchronization Sources and Rules step, configure basic information such as the name of the data sync solution.

   In the **Basic Configuration** section, set the parameters that are described in the following table.



Basic configuration

* Scheme name: [redacted]

Description:

Destination task storage location: ☑ Automatically establish workflow

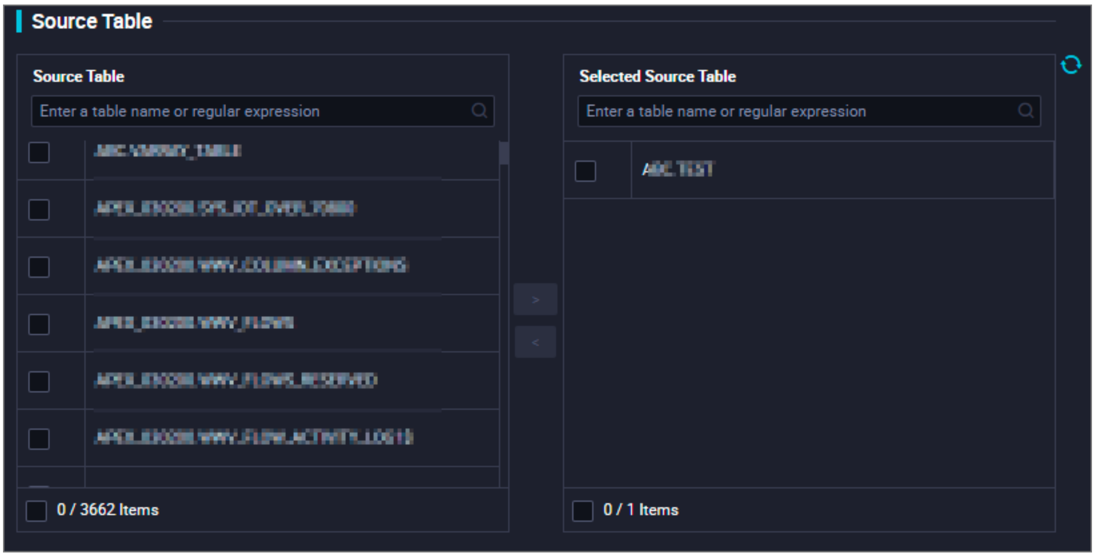
Parameter	Description
Solution Name	The name of the sync solution. The name can be a maximum of 50 characters in length.
Description	The description of the sync solution. The description can be a maximum of 50 characters in length.
Location	If you select Automatic Workflow Creation, DataWorks automatically creates a workflow named in the format of clone_database_Source name+to+Destination name. All sync nodes generated by the sync solution are placed in the **Data Integration** folder of this workflow. If you clear **Automatic Workflow Creation**, select a directory from the **Select Location** drop-down list. All sync nodes generated by the data sync solution are placed in the specified directory.

5. Select a source data source and configure sync rules.

   i. In the **Data Source** section, specify the **Type** and **Data source** parameters.

   Note You can set the Type parameter only to MySQL, Oracle, or PolarDB.

ii. In the **Source Table** section, select the tables whose data you want to synchronize from the **Source Table** list. Then, click the > icon to add the tables to the **Selected Source Table** list.



Source Table

Source Table

Enter a table name or regular expression

□ [redacted]

□ 0 / 3662 Items

Selected Source Table

Enter a table name or regular expression

□ [redacted]

□ 0 / 1 Items

The Source Table section displays all the tables in the source. You can select all or specific tables.

> **Notice** If a selected table does not have a primary key, the table cannot be synchronized in real time.

iii. In the **Mapping Rules for Table Names** section, click **Add rule** to select a rule.

Supported options are **Conversion Rule for Table Name** and **Rule for Destination Table name**.

- **Conversion Rule for Table Name**: the rule that is used to convert the names of source tables to those of destination tables.
- **Rule for Destination Table name**: the rule that is used to add a prefix or a suffix to the converted names of destination tables.

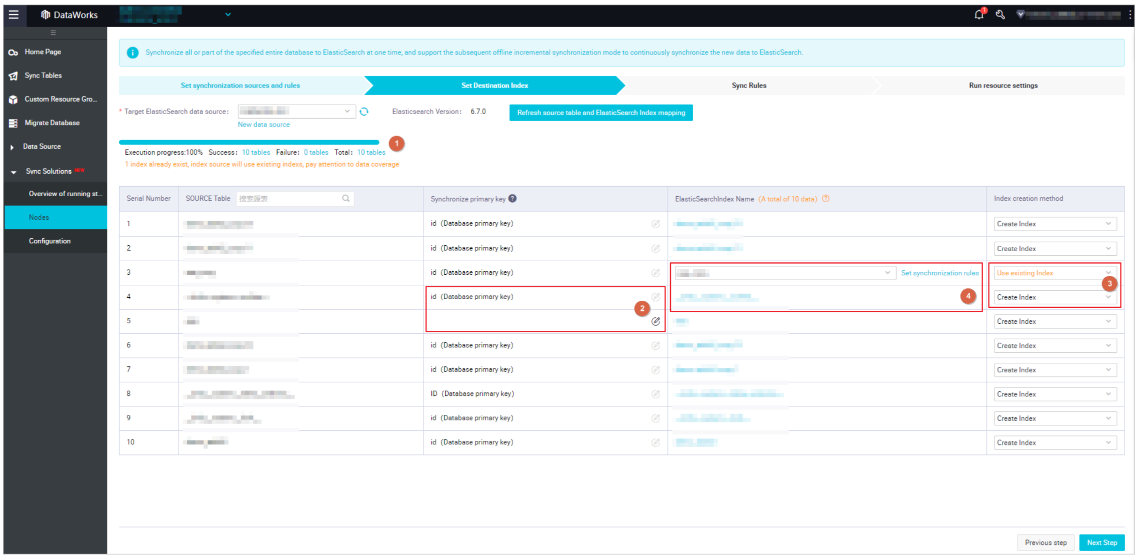
iv. Click **Next Step**.

6. Select the destination data source and configure the formats for the destination tables.

i. In the **Set Destination Table** step, specify **Destination** and **Schema**, and specify whether to enable **Table name case sensitive**.

ii. Click **Refresh source table and Hologres Table mapping** to configure the mappings between the source tables and destination Hologres tables.

iii. View the mapping progress, source tables, and mapped destination tables.



Synchronize all or part of the specified entire database to ElasticSearch at one time, and support the subsequent offline incremental synchronization mode to continuously synchronize the new data to ElasticSearch.

* Target ElasticSearch data source: [redacted]

New data source

Elasticsearch Version: 6.7.0

Refresh source table and ElasticSearch Index mapping

Execution progress:100% Success: 10 tables Failure: 0 tables Total: 10 tables

1 index already exist, index source will use existing indexs, pay attention to data coverage

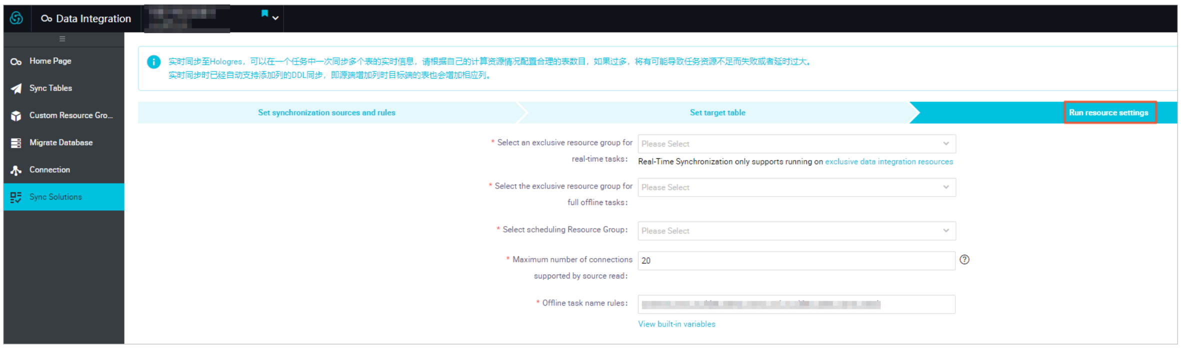
Serial Number	SOURCE Table 搜索源表	Synchronize primary key	ElasticSearchIndex Name (A total of 10 data)	Index creation method
1	[redacted]	id (Database primary key)	[redacted]	Create Index
2	[redacted]	id (Database primary key)	[redacted]	Create Index
3	[redacted]	id (Database primary key)	[redacted] Set synchronization rules	Use existing Index
4	[redacted]	id (Database primary key)	[redacted]	Create Index
5	[redacted]		[redacted]	Create Index
6	[redacted]	id (Database primary key)	[redacted]	Create Index
7	[redacted]	id (Database primary key)	[redacted]	Create Index
8	[redacted]	ID (Database primary key)	[redacted]	Create Index
9	[redacted]	id (Database primary key)	[redacted]	Create Index
10	[redacted]	id (Database primary key)	[redacted]	Create Index

Serial number	Description
1	The progress of mapping the source tables to destination tables. Note The mapping may require an extended period of time if you want to synchronize data from a large number of tables.
2	If a source table does not have a primary key, an error message appears to remind you that the current source table does not have a primary key and cannot be synchronized. The synchronization can be performed if one of the selected source tables has a primary key. Source tables without primary keys are ignored during the synchronization.
3	The source of the destination table. Valid values: **Create Table** and **Use Existing Table**.
4	The name of the destination table. The information that appears here varies based on the value that you selected from the drop-down list in the **Table creation method** column. ■ If you set the **Table creation method** parameter to **Use Existing Table**, the names of existing Hologres tables are automatically displayed in the drop-down list of the **Hologres Table name** column. You can select the table name that you want to use from the drop-down list. ■ If you set the **Table creation method** to **Create Table**, the name of the destination table that is automatically created appears. To view and modify the SQL statements that are used to create a table, click the name of the table.

iv. Click **Next Step**.

7. Configure the resources required by the data sync solution.

   In the **Set Resources for Solution Running** step, set the parameters that are described in the following table.



实时同步至Hologres，可以在一个任务中一次同步多个表的实时信息，请根据自己的计算资源情况配置合理的表数目，如果过多，将有可能导致任务资源不足而失败或者延时过大。
实时同步时已经自动支持添加列的DDL同步，即源端增加列时目标端的表也会增加相应列。

Set synchronization sources and rules > Set target table > Run resource settings

* Select an exclusive resource group for real-time tasks: Please Select
Real-Time Synchronization only supports running on exclusive data integration resources

* Select the exclusive resource group for full offline tasks: Please Select

* Select scheduling Resource Group: Please Select

* Maximum number of connections supported by source read: 20

* Offline task name rules: [redacted]
View built-in variables

Parameter	Description
Select an exclusive resource group for real-time tasks	The exclusive resource group that is used to run the real-time sync node and batch sync node generated by the data sync solution. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources.
Resource Groups for Full Batch Sync Nodes	
Select scheduling Resource Group	The resource group for scheduling that is used to run the nodes generated by the batch sync solution.
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source.
Offline task name rules	The name of the batch sync node that is used to synchronize the full data of the source database. After a sync solution is configured, DataWorks first runs a batch sync node to synchronize full data, and then runs a real-time sync node to synchronize incremental data.

8. Click **Complete Configuration**. The sync solution is configured.

Run the sync solution

On the **Tasks** page, find the created data sync solution and click **Submit and Run** in the Operation column to run the data sync solution.

If the execution of the solution fails, you can view the error message and troubleshoot the issues based on the following answers to some commonly asked questions:

- The system displays the following error message for a real-time synchronization node: "com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX." What do I do?
- The system displays the following error message for a real-time data synchronization node: "com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation." What do I do?
- The system displays the following error message for a real-time data synchronization node: "com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first." What do I do?
- The system displays the following error message for the batch synchronization node: "com.alibaba.datax.common.exception.DataXException: Code:[HoloWriter-02], Description:[Invalid config parameter in your configuration.]. - Field _log_file_name_offset_ not allow null but not present in user configured columns." What do I do?

View the status and result of the data sync nodes

- On the **Tasks** page, find the solution that is run and choose More > **Execution details** in the Operation column. Then, you can view the execution details of all nodes.
- Find a node whose execution details you want to view and click **Execution details** in the Status column. In the message that appears, click the provided link to go to the DataStudio page.

Manage the data sync solution

- View or edit the data sync solution.

  On the **Tasks** page, find the solution that you want to view or edit, and choose More > **View Configuration** in the Operation column.

  > Note You can click **View Configuration** to modify the sync solution only if the solution is in the **Not Running** state. If you click View Configuration in the Operation column that corresponds to a data sync solution in another state, you can view only the information about that data sync solution.

- Delete the data sync solution.

  Find the solution that you want to delete and choose More > **Delete** in the Operation column. In the **Delete** message, click **OK**.

  > Note After you click OK, only the configuration record of the data sync solution is deleted. The generated sync nodes and tables are not affected.

4.5.9. Add or remove source tables to or from a synchronization solution that is running

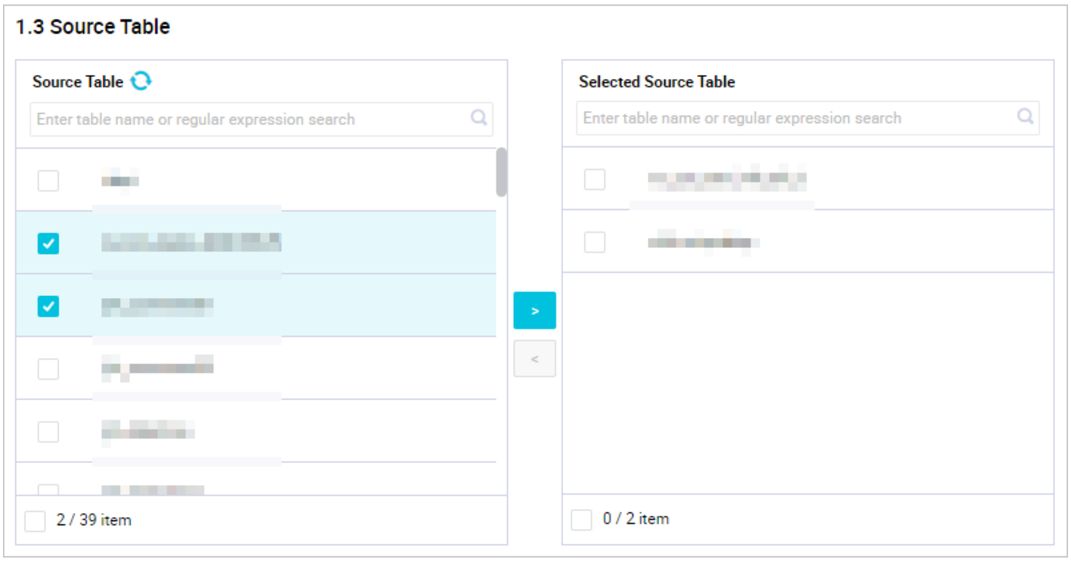
For a solution used to synchronize data to Hologres, you can add or remove source tables when the solution is running. This topic describes how to add or remove source tables to or from a synchronization solution that is running.

Prerequisites

A synchronization solution used to synchronize data to Hologres is created and running. For more information, see Create and configure a sync solution.

Add source tables to a synchronization solution

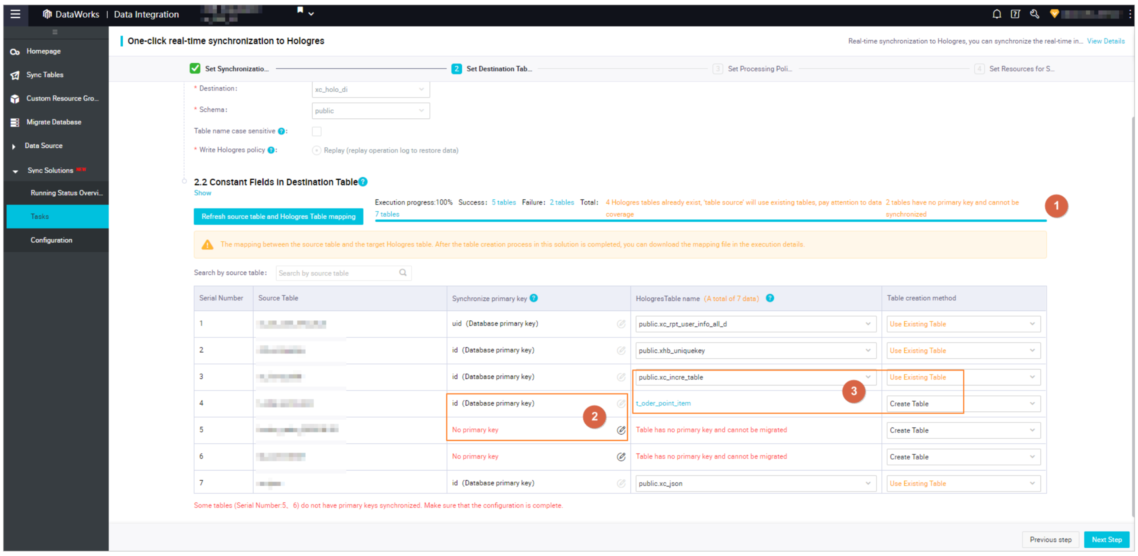
1.
2. On the **Tasks** page, find the desired solution and choose **More > Modify Configuration** to go to the solution configuration page.
3. Add source tables to the synchronization solution and update the mappings between the source tables and destination tables.
   i. In the **Source Table** section of the **Set Synchronization Sources and Rules** step, select the source tables that you want to add to the synchronization solution from the **Source Table** list and click the > icon to move the tables to the **Selected Source Table** list.


1.3 Source Table
Source Table
Enter table name or regular expression search
2 / 39 item
>
<
Selected Source Table
Enter table name or regular expression search
0 / 2 item

   ii. Click **Next Step**.

iii. Click **Refresh source table and Hologres Table mapping** in the **Set Destination Table** step to update the mappings between the source tables and destination Hologres tables.

iv. View the mapping progress, source tables, and mapped destination tables.



One-click real-time synchronization to Hologres

Real-time synchronization to Hologres, you can synchronize the real-time in... View Details

* Destination: xc_holo_di

* Schema: public

Table name case sensitive: □

* Write Hologres policy: ○ Replay (replay operation log to restore data)

2.2 Constant Fields In Destination Table

Show

Refresh source table and Hologres Table mapping

Execution progress:100% Success: 5 tables Failure: 2 tables Total: 7 tables 4 Hologres tables already exist, 'table source' will use existing tables, pay attention to data coverage 2 tables have no primary key and cannot be synchronized

The mapping between the source table and the target Hologres table. After the table creation process in this solution is completed, you can download the mapping file in the execution details.

Search by source table: Search by source table

Serial Number	Source Table	Synchronize primary key	HologresTable name (A total of 7 data)	Table creation method
1	[redacted]	uid (Database primary key)	public.xc_rpt_user_info_all_d	Use Existing Table
2	[redacted]	id (Database primary key)	public.xhb_uniquekey	Use Existing Table
3	[redacted]	id (Database primary key)	public.xc_incre_table	Use Existing Table
4	[redacted]	id (Database primary key)	t_oder_point_item	Create Table
5	[redacted]	No primary key	Table has no primary key and cannot be migrated	Create Table
6	[redacted]	No primary key	Table has no primary key and cannot be migrated	Create Table
7	[redacted]	id (Database primary key)	public.xc_json	Use Existing Table

Some tables (Serial Number:5、6) do not have primary keys synchronized. Make sure that the configuration is complete.

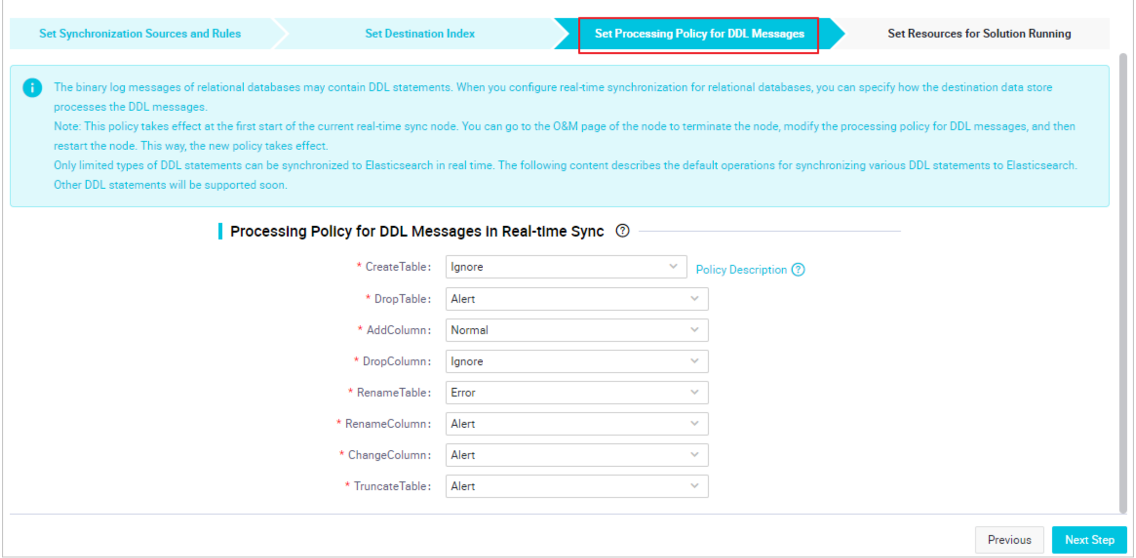
No.	Description
1	The progress of mapping the source tables to destination tables. **Note** The mapping may require a long period of time if you want to synchronize data from a large number of tables.
2	■ If the tables in the source database contain primary keys, the system removes duplicate data based on the primary keys during the synchronization. ■ If the tables in the source database do not contain primary keys, you can click the ✎ icon to customize primary keys. You can use one field or a combination of several fields as the primary keys of the tables. This way, the system removes duplicate data based on the primary keys during the synchronization.
3	

4. Click **Next Step**.

5. Configure rules to process data definition language (DDL) messages.

   Sources, such as MySQL, may contain DDL messages. You can modify the configured processing rules for different DDL messages based on your business requirements in the **Set Processing Policy for DDL Messages** step.

i. Configure parameters in the **Processing Policy for DDL Messages in Real-time Sync** section.



Set Synchronization Sources and Rules
Set Destination Index
Set Processing Policy for DDL Messages
Set Resources for Solution Running

The binary log messages of relational databases may contain DDL statements. When you configure real-time synchronization for relational databases, you can specify how the destination data store processes the DDL messages.
Note: This policy takes effect at the first start of the current real-time sync node. You can go to the O&M page of the node to terminate the node, modify the processing policy for DDL messages, and then restart the node. This way, the new policy takes effect.
Only limited types of DDL statements can be synchronized to Elasticsearch in real time. The following content describes the default operations for synchronizing various DDL statements to Elasticsearch. Other DDL statements will be supported soon.

Processing Policy for DDL Messages in Real-time Sync

- * CreateTable: Ignore
- * DropTable: Alert
- * AddColumn: Normal
- * DropColumn: Ignore
- * RenameTable: Error
- * RenameColumn: Alert
- * ChangeColumn: Alert
- * TruncateTable: Alert

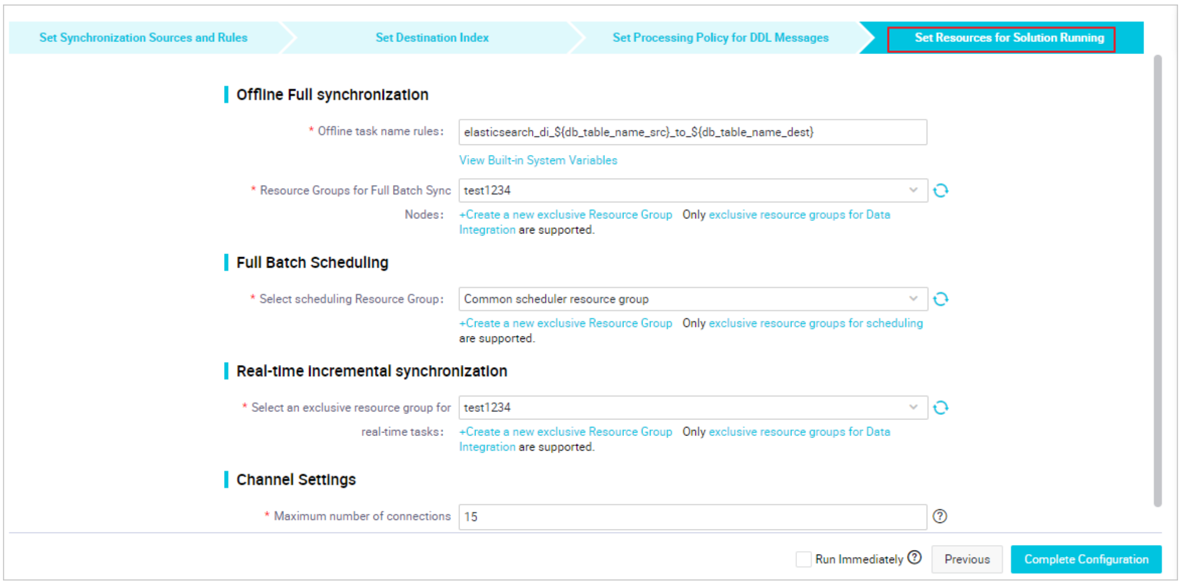
The following table describes the processing rules for different DDL messages.

DDL message	Rule
CreateTable	DataWorks processes a DDL message of the related type based on the following rules after it receives the message: ■ **Normal**: sends the message to the destination. Then, the destination processes the message. Each destination may process DDL messages based on its own business logic. If you select Normal for CreateTable, DataWorks only forwards the messages. ■ **Ignore**: ignores the message and does not send it to the destination. ■ **Alert**: ignores the message and records the alert in real-time synchronization logs. In addition, the alert contains information about the reason indicating that a message is ignored because of a running error. ■ **Error**: returns an error when the real-time sync solution is running and terminates the real-time sync solution.
DropTable	
AddColumn	
DropColumn	
RenameTable	
RenameColumn	
ChangeColumn	
TruncateTable	

ii. Click **Next Step**.

6. Configure the resources required by the sync solution.

In the **Set Resources for Solution Running** step, set the parameters as required.


Set Synchronization Sources and Rules
Set Destination Index
Set Processing Policy for DDL Messages
Set Resources for Solution Running

Offline Full synchronization

* Offline task name rules: elasticsearch_di_${db_table_name_src}_to_${db_table_name_dest}

View Built-in System Variables

* Resource Groups for Full Batch Sync Nodes: test1234

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

Full Batch Scheduling

* Select scheduling Resource Group: Common scheduler resource group

+Create a new exclusive Resource Group Only exclusive resource groups for scheduling are supported.

Real-time Incremental synchronization

* Select an exclusive resource group for real-time tasks: test1234

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

Channel Settings

* Maximum number of connections 15

- Offline Sync

Parameter	Description
Offline task name rules	The name of the batch sync node that is used to synchronize the full data of the source. After a sync solution is created, DataWorks first generates a batch sync node to synchronize full data, and then generates real-time sync nodes to synchronize incremental data.
Resource Groups for Full Batch Sync Nodes	The exclusive resource group for Data Integration that is used to run the batch sync node. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Scheduling Settings

Parameter	Description
Select scheduling Resource Group	The resource group for scheduling that is used to run the nodes. Only exclusive resource groups for Data Integration can be used to run sync solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Incremental Sync

Parameter	Description

Parameter	Description
Resource Groups for Incremental Batch Sync Nodes	The exclusive resource group that is used to run the real-time sync nodes. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. ⓘ **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- **Channel Settings**

Parameter	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: **20**.

7. Click **Complete Configuration** to return to the **Tasks** page.
8. Find the solution to which you added source tables and choose **More > Submit and Run** in the **Operation** column. In the **Submit and Run** message, click **OK** to run the solution.

   After you submit and run the solution to which you added source tables, the system compares the source tables in the original solution with the source tables in the new solution. If new source tables are detected, the system performs the process of adding the source tables.

   > ⓘ **Note** After you add source tables to the synchronization solution at a specific point in time, the system starts to load data to these newly added source tables at this point in time. After the data loading ends, the system starts to synchronize the data in these source tables to the destination. For example, your synchronization solution starts to run at 08:00 and is still running at 09:00. You add a source table to the solution at 09:00. Then, the system starts to load data to the table from 09:00, and the loading is ended at 10:00. In this case, the system stops the real-time synchronization nodes that are running and starts to synchronize the data that is generated from 09:00 to 10:00 in the newly added source table to the destination Hologres table. The addition of source tables to a synchronization solution that is running can ensure only the consistency between data before and after the synchronization.

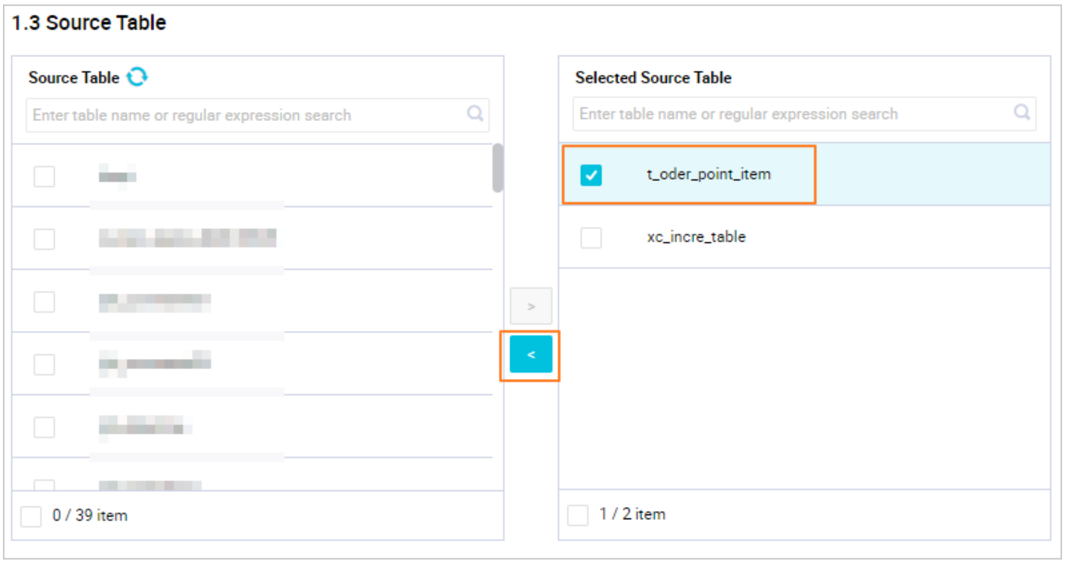
9. View the addition details of the source tables.
   i. On the **Tasks** page, find the synchronization solution to which you added source tables and click **Execution details** in the **Operation** column to go to the details page of the synchronization solution.
   ii. In the **Steps** section, find the **Show Added or Removed Source Tables** node and click **Execution details** in the Status column.

   If the state of the **Show Added or Removed Tables** node is **Succeeded**, the new source tables are added for the synchronization solution.
   iii. View the new source tables that are added to the synchronization solution.

Remove source tables from the synchronization solution

1.
2. On the **Tasks** page, find the desired solution and choose **More > Modify Configuration** to go to the solution configuration page.
3. Remove source tables from the synchronization solution and update the mappings between the remaining source tables and destination tables.

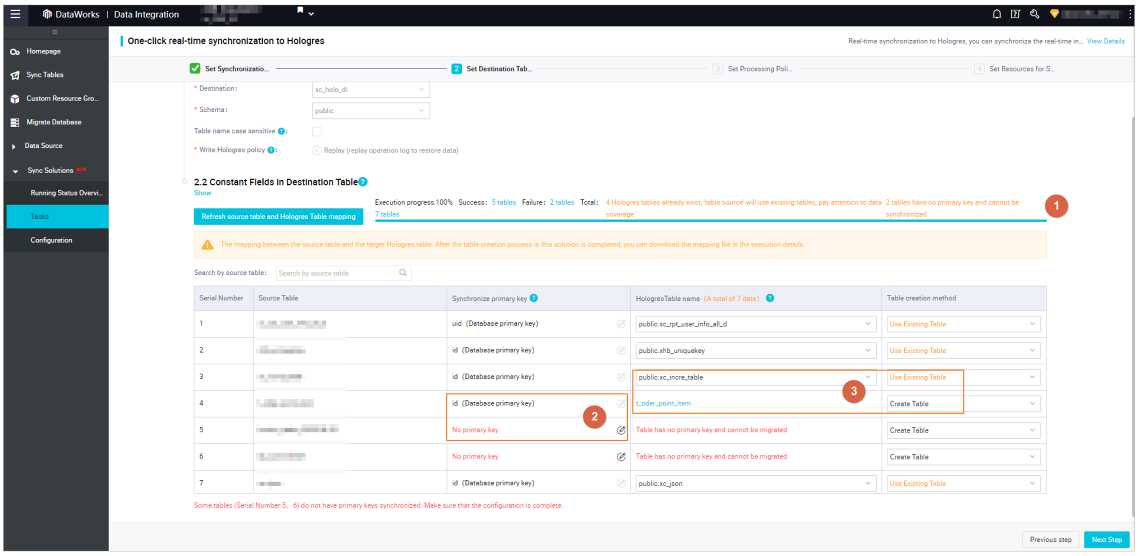
i. In the **Source Table** section of the **Set Synchronization Sources and Rules** step, select the source tables that you want to remove from the synchronization solution in the **Selected Source Table** list and click the < icon to move the tables back to the **Source Table** list.



ii. Click **Next Step**.

iii. Click **Refresh source table and Hologres Table mapping** in the **Set Destination Table** step to update the mappings between the source tables and destination Hologres tables.

iv. View the mapping progress, source tables, and mapped destination tables.



One-click real-time synchronization to Hologres

Real-time synchronization to Hologres, you can synchronize the real-time in... View Details

* Destination: xc_holo_di

* Schema: public

Table name case sensitive: □

* Write Hologres policy: ○ Replay (replay operation log to restore data)

2.2 Constant Fields in Destination Table

Show

Refresh source table and Hologres Table mapping

Execution progress:100% Success: 5 tables Failure: 2 tables Total: 7 tables 4 Hologres tables already exist, 'table source' will use existing tables, pay attention to data coverage 2 tables have no primary key and cannot be synchronized

The mapping between the source table and the target Hologres table. After the table creation process in this solution is completed, you can download the mapping file in the execution details.

Search by source table: Search by source table

Serial Number	Source Table	Synchronize primary key	HologresTable name (A total of 7 data)	Table creation method
1	[redacted]	uid (Database primary key)	public.xc_rpt_user_info_all_d	Use Existing Table
2	[redacted]	id (Database primary key)	public.xhb_uniquekey	Use Existing Table
3	[redacted]	id (Database primary key)	public.xc_incre_table	Use Existing Table
4	[redacted]	id (Database primary key)	t_oder_point_item	Create Table
5	[redacted]	No primary key	Table has no primary key and cannot be migrated	Create Table
6	[redacted]	No primary key	Table has no primary key and cannot be migrated	Create Table
7	[redacted]	id (Database primary key)	public.xc_json	Use Existing Table

Some tables (Serial Number:5、6) do not have primary keys synchronized. Make sure that the configuration is complete.

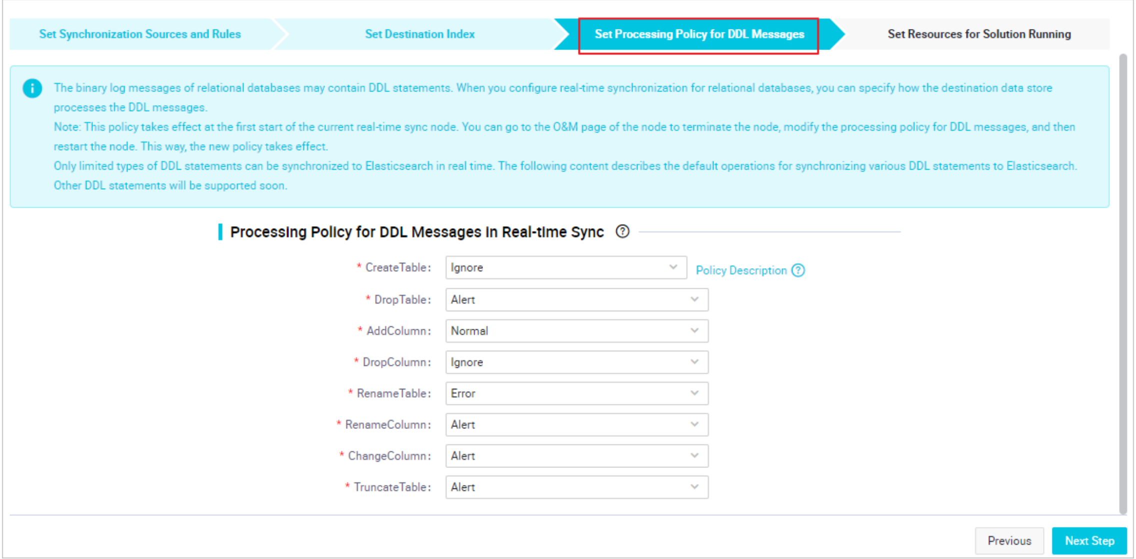
No.	Description
1	The progress of mapping the source tables to destination tables. Note The mapping may require a long period of time if you want to synchronize data from a large number of tables.
2	■ If the tables in the source database contain primary keys, the system removes duplicate data based on the primary keys during the synchronization. ■ If the tables in the source database do not contain primary keys, you can click the ✎ icon to customize primary keys. You can use one field or a combination of several fields as the primary keys of the tables. This way, the system removes duplicate data based on the primary keys during the synchronization.
3	

4. Click **Next Step**.

5. Configure rules to process data definition language (DDL) messages.

   Sources, such as MySQL, may contain DDL messages. You can modify the configured processing rules for different DDL messages based on your business requirements in the **Set Processing Policy for DDL Messages** step.

i. Configure parameters in the **Processing Policy for DDL Messages in Real-time Sync** section.



Set Synchronization Sources and Rules
Set Destination Index
Set Processing Policy for DDL Messages
Set Resources for Solution Running

The binary log messages of relational databases may contain DDL statements. When you configure real-time synchronization for relational databases, you can specify how the destination data store processes the DDL messages.
Note: This policy takes effect at the first start of the current real-time sync node. You can go to the O&M page of the node to terminate the node, modify the processing policy for DDL messages, and then restart the node. This way, the new policy takes effect.
Only limited types of DDL statements can be synchronized to Elasticsearch in real time. The following content describes the default operations for synchronizing various DDL statements to Elasticsearch. Other DDL statements will be supported soon.

Processing Policy for DDL Messages in Real-time Sync

* CreateTable: Ignore
* DropTable: Alert
* AddColumn: Normal
* DropColumn: Ignore
* RenameTable: Error
* RenameColumn: Alert
* ChangeColumn: Alert
* TruncateTable: Alert

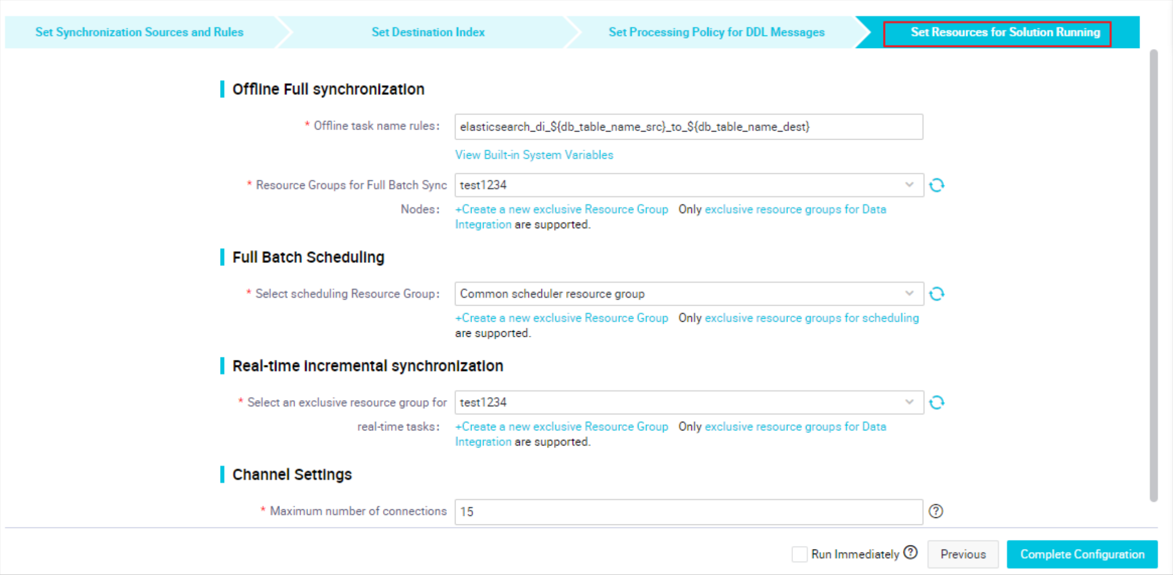
The following table describes the processing rules for different DDL messages.

DDL message	Rule
CreateTable	DataWorks processes a DDL message of the related type based on the following rules after it receives the message: ■ **Normal**: sends the message to the destination. Then, the destination processes the message. Each destination may process DDL messages based on its own business logic. If you select Normal for CreateTable, DataWorks only forwards the messages. ■ **Ignore**: ignores the message and does not send it to the destination. ■ **Alert**: ignores the message and records the alert in real-time synchronization logs. In addition, the alert contains information about the reason indicating that a message is ignored because of a running error. ■ **Error**: returns an error when the real-time sync solution is running and terminates the real-time sync solution.
DropTable	
AddColumn	
DropColumn	
RenameTable	
RenameColumn	
ChangeColumn	
TruncateTable	

ii. Click **Next Step**.

6. Configure the resources required by the sync solution.

   In the **Set Resources for Solution Running** step, set the parameters as required.



Set Synchronization Sources and Rules | Set Destination Index | Set Processing Policy for DDL Messages | Set Resources for Solution Running

Offline Full synchronization

* Offline task name rules: elasticsearch_di_${db_table_name_src}_to_${db_table_name_dest}

View Built-in System Variables

* Resource Groups for Full Batch Sync Nodes: test1234

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

Full Batch Scheduling

* Select scheduling Resource Group: Common scheduler resource group

+Create a new exclusive Resource Group Only exclusive resource groups for scheduling are supported.

Real-time Incremental synchronization

* Select an exclusive resource group for real-time tasks: test1234

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

Channel Settings

* Maximum number of connections 15

□ Run Immediately ⓘ Previous Complete Configuration

- Offline Sync

Parameter	Description
Offline task name rules	The name of the batch sync node that is used to synchronize the full data of the source. After a sync solution is created, DataWorks first generates a batch sync node to synchronize full data, and then generates real-time sync nodes to synchronize incremental data.
Resource Groups for Full Batch Sync Nodes	The exclusive resource group for Data Integration that is used to run the batch sync node. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. ⓘ **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Scheduling Settings

Parameter	Description
Select scheduling Resource Group	The resource group for scheduling that is used to run the nodes. Only exclusive resource groups for Data Integration can be used to run sync solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. ⓘ **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Incremental Sync

Parameter	Description

Parameter	Description
Resource Groups for Incremental Batch Sync Nodes	The exclusive resource group that is used to run the real-time sync nodes. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- **Channel Settings**

Parameter	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: **20**.

7. Click **Complete Configuration** to return to the **Tasks** page.
8. Find the solution from which you removed source tables and choose **More > Submit and Run** in the **Operation** column. In the **Submit and Run** message, click **OK** to run the solution.

   If you remove source tables from a synchronization solution that is running, the source tables are also removed from real-time synchronization nodes generated by the synchronization solution. After you submit and run the synchronization solution from which you removed source tables, the system continues to synchronize data at the time when the synchronization solution starts to be rerun.
9. View the removal details of the source tables.
   i.
   ii. In the **Steps** section, find the **Show Added or Removed Source Tables** node and click **Execution details** in the Status column.

   If the state of the **Show Added or Removed Source Tables** node is **Succeeded**, the source tables are removed from the synchronization solution.
   iii. View the source tables that are removed from the synchronization solution.

4.5.10. FAQ

This topic provides answers to some frequently asked questions about data synchronization to Hologres.

- What do I do if the PolarDB data source cannot be connected?
- The system displays the following error message for a real-time synchronization node: "com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX." What do I do?
- The system displays the following error message for a real-time data synchronization node: "com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation." What do I do?
- The system displays the following error message for a real-time data synchronization node: "com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first." What do I do?
- The system displays the following error message for the batch synchronization node: "com.alibaba.datax.common.exception.DataXException: Code:[HoloWriter-02], Description:[Invalid config parameter in your configuration.]. - Field _log_file_name_offset_ not allow null but not present in user configured columns." What do I do?
- The system displays the following error message for the batch synchronization node: "errorCode:NoSuchTopic, errorMessage:The specified topic name does not exist." What do I do?

What do I do if the PolarDB data source cannot be connected?

- Problem description: The connectivity test fails when I add a PolarDB data source.
- Solution: Set Data source type to Connection string mode and check the whitelist configuration of the PolarDB cluster and the virtual private cloud (VPC) configuration of your exclusive resource group.

What do I do if the connectivity test fails for the Oracle data source?

- Problem description: The connectivity test fails when I add an Oracle data source.
- Solution: Set Data source type to Connection string mode and check the whitelist configuration of the PolarDB cluster and the virtual private cloud (VPC) configuration of your exclusive resource group.

What do I do if the connectivity test fails for the MySQL data source?

- Problem description: The connectivity test fails when I add a MySQL data source.
- Solution: Set Data source type to Connection string mode and check the whitelist configuration of the PolarDB cluster and the virtual private cloud (VPC) configuration of your exclusive resource group.

The system displays the following error message for a real-time synchronization node: "com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX." What do I do?

- Problem description: The real-time synchronization node fails to run, and the system displays the error message "`com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX`."
- Cause: The binary logging feature is disabled for the PolarDB data source.
- Solution: Enable the binary logging feature for the PolarDB data source. For more information, see Configure a data source (PolarDB). Change one or more data records and change the start time for running the real-time data synchronization node to the current time.

The system displays the following error message for a real-time data synchronization node: "com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation." What do I do?

- Problem description: The real-time synchronization node fails to run, and the system displays the error message "`com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation`."
- Cause: The account used to synchronize data is not authorized to access the PolarDB data source, or the PolarDB database connected is not deployed on the PolarDB Writer node.
- Solution: Authorize the account to access the PolarDB data source. For more information, see Configure a data source (PolarDB). You can also check whether the PolarDB database connected is deployed on the PolarDB Writer node. When a real-time data synchronization node is run, the system cannot capture data from the PolarDB Reader nodes of the PolarDB cluster.

The system displays the following error message for a real-time data synchronization node: "com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first." What do I do?

- Problem description: The real-time synchronization node fails to run, and the system displays the error message "`com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first`."
- Cause: The loose_polar_log_bin parameter is not specified for the PolarDB data source.
- Solution: Specify the loose_polar_log_bin parameter. For more information, see Configure a data source (PolarDB).

The system displays the following error message for the batch synchronization node: "com.alibaba.datax.common.exception.DataXException: Code:[HoloWriter-02], Description:[Invalid config parameter in your configuration.]. - Field _log_file_name_offset_ not allow null but not present in user configured columns." What do I do?

- Problem description: The batch synchronization node fails to run, and the system displays the error message " `com.alibaba.datax.common.exception.DataXException: Code:[HoloWriter-02], Description:[Invalid config parameter in your configuration.]. - Field _log_file_name_offset_ not allow null but not present in user configured columns` ."
- Cause: The engine-type plug-in of DataWorks for batch synchronization is not upgraded to the latest version.
- Solution: Submit a ticket to upgrade your plug-in.

The system displays the following error message for the batch synchronization node: "errorCode:NoSuchTopic, errorMessage:The specified topic name does not exist." What do I do?

- Problem description: The batch synchronization node fails to run, and the system displays the error message " `errorCode:NoSuchTopic, errorMessage:The specified topic name does not exist.` "
- Causes:
  - The destination Hologres table used for data synchronization does not exist.
  - The data synchronization node synchronizes data of the source table to a Hologres external table. The Hologres Writer node cannot write data to Hologres external tables.
- Solution: Use a Hologres internal table as the destination table for data synchronization. If the destination Hologres table does not exist, set Table creation method to **Create Table** when you configure a data synchronization node. Then, the destination Hologres table is automatically created. For more information, see Configure and view a data synchronization solution.

4.6. Synchronize data to AnalyticDB for MySQL V3.0

4.6.1. Plan and configure resources

If you use synchronization solutions of DataWorks to synchronize data, you can use only exclusive resource groups for Data Integration to run Data Integration nodes. However, you can select a shared resource group or an exclusive resource group for scheduling to use scheduling resources based on your business requirements. This topic describes the resources that are used for data synchronization and how to configure the resources.

Context

- Resource planning and preparation

  When you synchronize data, Data Integration nodes are run based on resource groups for Data Integration and resource groups for scheduling. You can use only exclusive resource groups for Data Integration to run Data Integration nodes. Before you synchronize data, you must purchase an exclusive resource group for Data Integration and add this exclusive resource group to your DataWorks workspace.

  For more information about exclusive resource groups for Data Integration, see Exclusive resources for Data Integration.

- Network connectivity

  An exclusive resource group for Data Integration is essentially a group of Elastic Compute Service (ECS) instances. After you purchase such an exclusive resource group, it is isolated from other services. You must associate the resource group with a virtual private cloud (VPC) to ensure network connectivity between the resource group and data sources during subsequent data synchronization.

What's next

After you plan and configure resources, you can configure data sources. You must connect the exclusive resource group for Data Integration to the source and destination. You must also create an account and grant the required permissions to the account. This account is used to access the source and destination. The preceding operations help you create a synchronization node. You can synchronize data to an AnalyticDB for MySQL V3.0 data source only from a PolarDB or MySQL data source. You can select a PolarDB or MySQL data source based on your business requirements. For more information about how to configure a data source, see Configure data sources for data synchronization from PolarDB, Configure data sources for data synchronization from ApsaraDB for OceanBase, or Configure data sources for data synchronization from MySQL.

4.6.2. Configure data sources for data synchronization from PolarDB

Before you use DataWorks to synchronize data from a PolarDB data source to AnalyticDB for MySQL V3.0, you need to configure data sources based on the operations in this topic. The configurations of data sources involve network environments, whitelists, and permissions.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

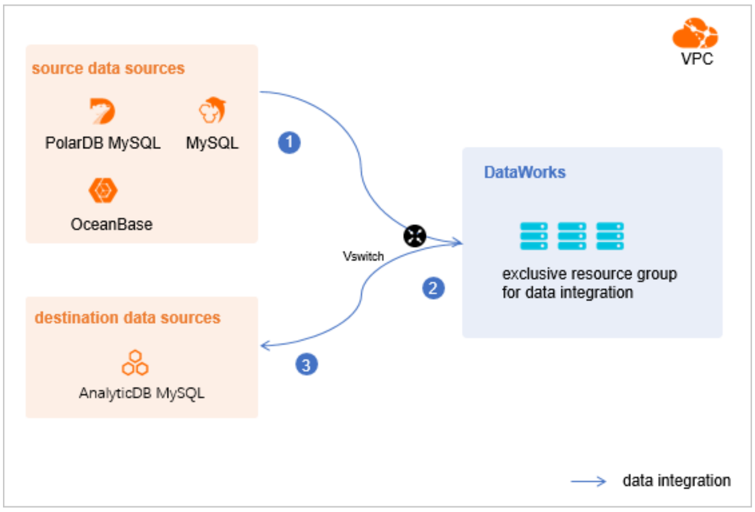
- Prepare data sources: A source PolarDB for MySQL cluster and a destination AnalyticDB for MySQL V3.0 cluster are prepared. In this topic, a PolarDB for MySQL cluster is used as the source.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
    - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
    - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

  If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the vSwitch that is bound to the exclusive resource group for Data Integration during network configuration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source is a PolarDB for MySQL cluster, you must enable the binary logging feature for the cluster. PolarDB for MySQL is fully compatible with MySQL and uses high-level physical logs to replace binary logs. To facilitate the integration between PolarDB and the MySQL ecosystem, you can enable the binary logging feature for PolarDB clusters.

Limits

- Only PolarDB for MySQL clusters can be used as sources in data synchronization solutions. Other types of PolarDB data sources are not supported. In this topic, PolarDB indicates PolarDB for MySQL data sources.
- Only data stored on the primary node of a PolarDB for MySQL cluster can be synchronized.

Procedure

1. Configure a whitelist for the PolarDB for MySQL cluster.

   To add the CIDR block of the VPC where the exclusive resource group for Data Integration resides to a whitelist of the PolarDB for MySQL cluster, perform the following steps:

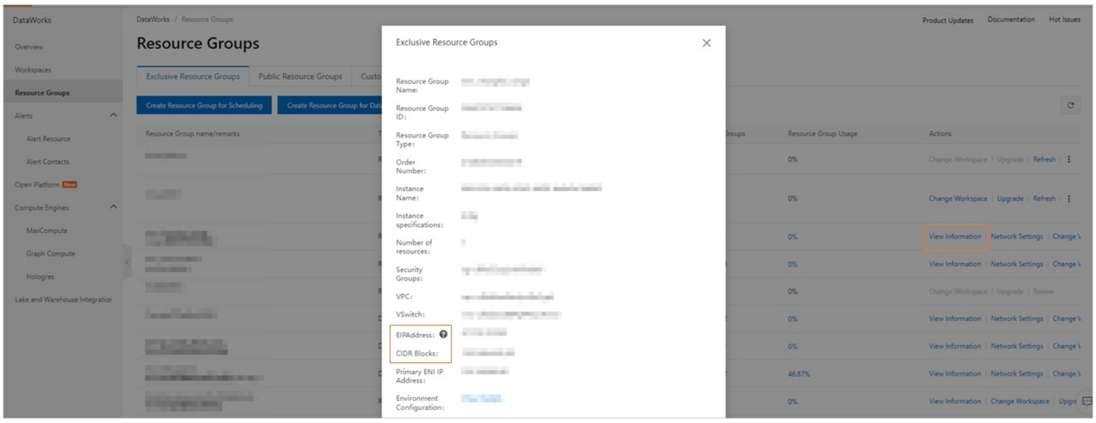
   i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

      a. Log on to the DataWorks console.

      b. In the left-side navigation pane, click **Resource Groups**.

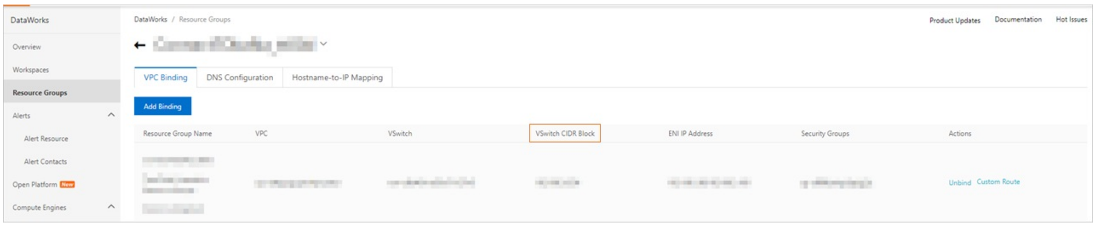
      c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

      d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.

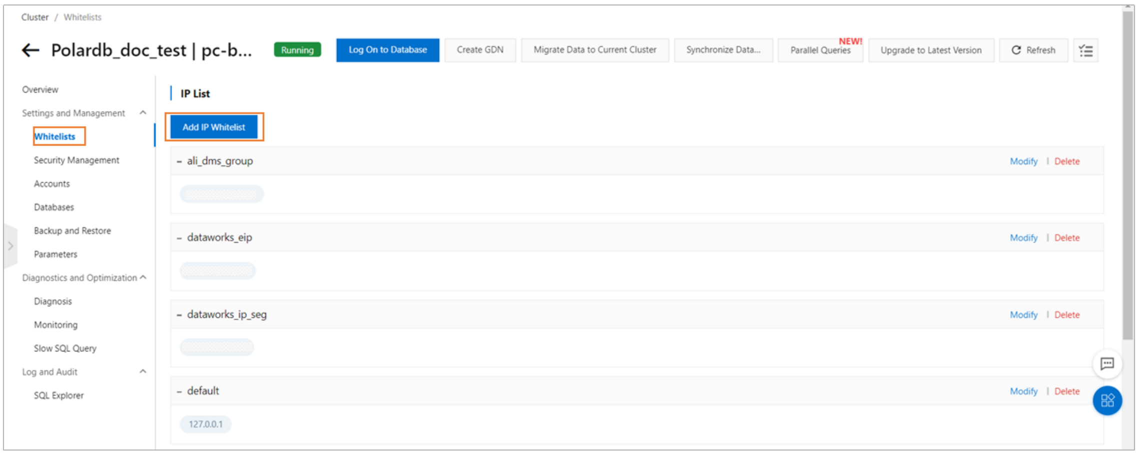


      e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

      f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR blocks recorded in the preceding steps to the whitelist of the PolarDB for MySQL cluster.



For more information, see Configure an IP whitelist.

2. Create an account and grant the required permissions to the account.

   You must create an account to log on to the database of the PolarDB for MySQL cluster. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

   i. Create an account.

      For more information, see Create a database account.

   ii. Grant the required permissions to the account.

      You can run the following command to grant the required permissions to the account, or you can directly assign the `SUPER` role to the account.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Account for data synchronization';
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%';
```

3. Enable the binary logging feature for the PolarDB for MySQL cluster.

   For more information, see Enable binary logging.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add data sources.

4.6.3. Configure data sources for data synchronization from MySQL

Before you use DataWorks to synchronize data from a MySQL data source to AnalyticDB for MySQL V3.0, you need to configure data sources based on the operations in this topic. The configurations of data sources involve network environments, whitelists, and permissions.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

- Prepare data sources: A source MySQL database and a destination AnalyticDB for MySQL V3.0 cluster are prepared.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.

- If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
- If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

- Execute the following statement to check the version of your MySQL database. You can use a `MySQL 5.X` or `MySQL 8.X` database.

```
select version();
```

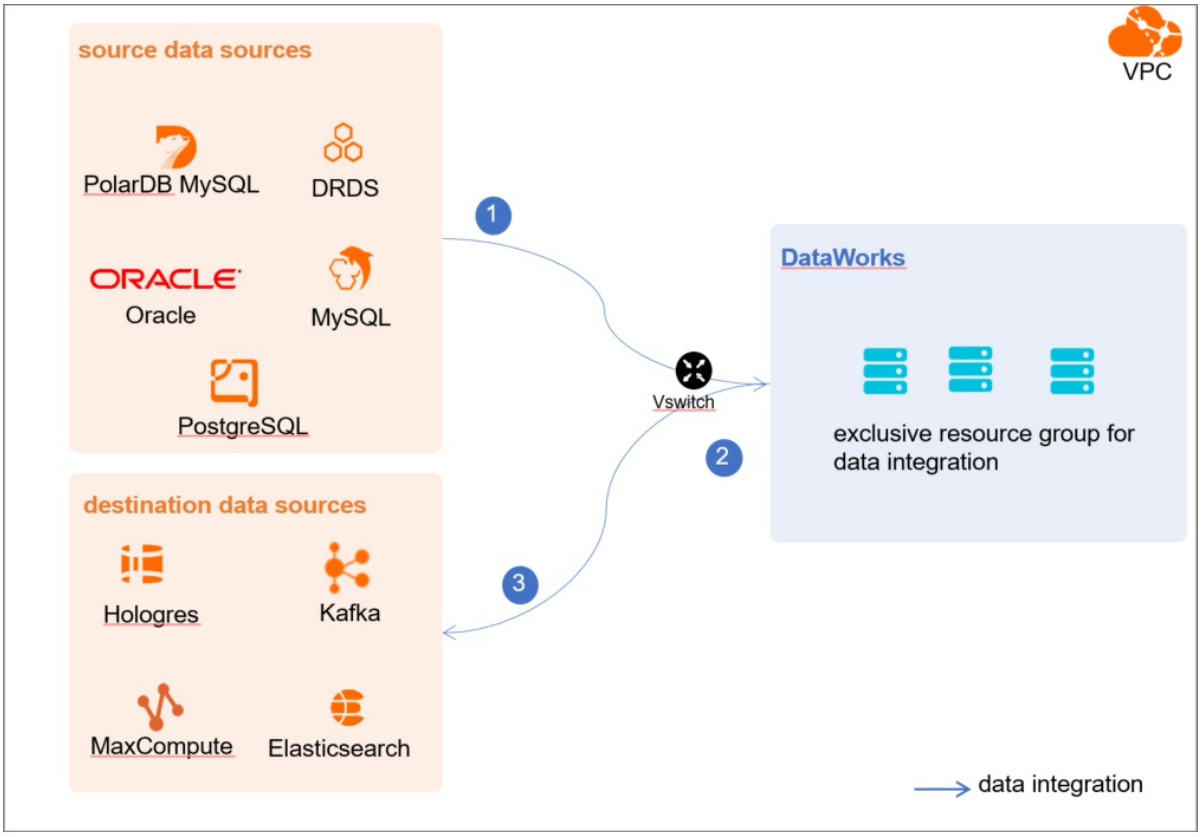
Note Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.X` or `V8.X`. PolarDB for MySQL is not supported. If the MySQL version of your ApsaraDB RDS for MySQL database is not `V5.X` or `V8.X`, use an ApsaraDB RDS for MySQL database whose MySQL version is `V5.X` or `V8.X`. Otherwise, the data synchronization node fails to run.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

  If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

If the source data source is a MySQL database, you must enable the binary logging feature. Binary logs record changes to all table schemas and modifications on table data. You can execute statements such as CREATE and ALTER to perform operations on table schemas. You can execute statements such as INSERT, UPDATE, and DELETE to perform operations on table data. You can use binary logs to view the change history of the database, back up incremental data and restore data in the database, and replicate data from the primary database to secondary databases.

Formats of binary logs:

- Statement: SQL statement-based replication. Binary logs in this format record the SQL statements that are executed to modify data entries.
- Row: row-based replication. Binary logs in this format record only modification details about data entries in rows.
- Mixed: replication in mixed mode. This mode combines the statement and row formats. In most cases, binary logs in the statement format are used to record the SQL statements that are executed to modify data entries, such as functions. If data replication from the primary database to secondary databases cannot be implemented by using binary logs in this format, switch to the row format. MySQL determines which format to use based on each SQL statement that is executed.

Limits

Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.X` or `V8.X`. PolarDB for MySQL is not supported.

Procedure

1. Configure a whitelist for the MySQL database.

   Add the CIDR block of the VPC in which the exclusive resource group resides to the whitelist of the MySQL database.

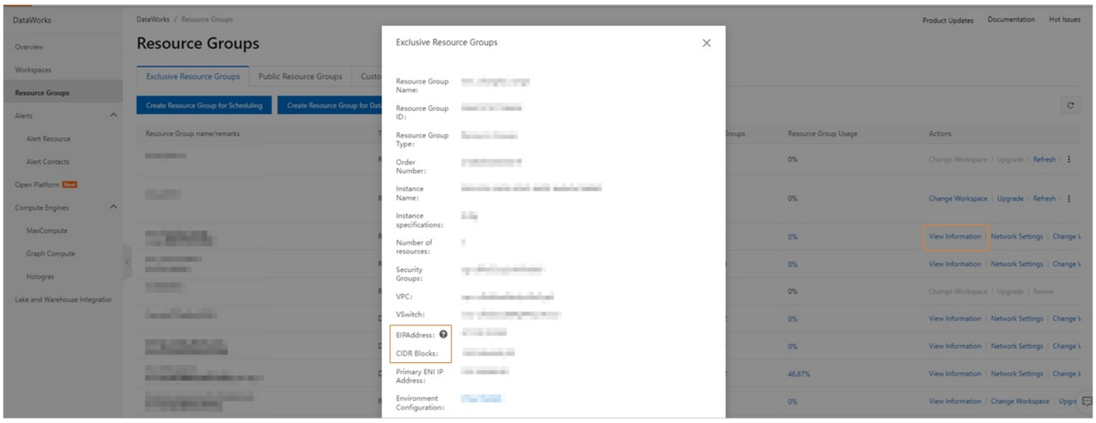
   i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

      a. Log on to the DataWorks console.

      b. In the left-side navigation pane, click **Resource Groups**.

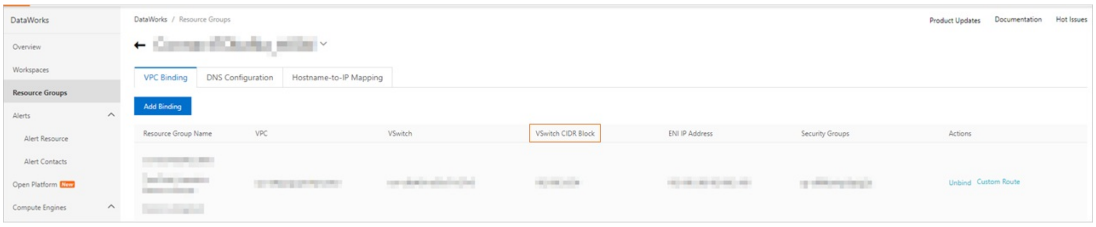
      c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

      d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



      e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

      f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



   ii. Add the EIP and CIDR block of the exclusive resource group to the whitelist of the MySQL database.

2. Create an account and grant permissions to the account.

   You must create an account to log on to the MySQL database. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

   i. Create an account.

      For more information, see Create an account to access a MySQL database.

   ii. Grant permissions to the account.

      You can run the following command to grant permissions to the account. Alternatively, you can grant the `SUPER` permission to the account. Replace `Account for data synchronization` with the created account when you execute the specific statement.

      ```
      -- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Password'; // Create an account that is used to synchronize data and set a password so that you can use the account and password to access the database from a host. % indicates a host.
      GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%'; // Grant the SELECT, REPLICATION SLAVE, and REPLICATION CLIENT permissions to the account.
      ```

      `*.*` indicates that the synchronization account is granted the preceding permissions on all tables in all databases. You can also grant the preceding permissions on specified tables in the specified database to the synchronization account. For example, to grant the account the preceding permissions on the *user* table in the *test* database, execute the following statement: `GRANT SELECT, REPLICATION CLIENT ON test.user TO 'Account for data synchronization'@'%';` .

      Note The `REPLICATION SLAVE` permission is a global permission. You cannot grant this permission on specified tables in the specified database to the synchronization account.

3. Enable the binary logging feature for the MySQL database.

   Perform the following steps to check whether the binary logging feature is enabled and to query the format of binary logs:

   - Execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_bin";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled.

   - If you use a secondary database to synchronize data, execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_slave_updates";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled for the secondary database.

   If the returned result is different from the preceding result, follow the instructions described in the *MySQL documentation* to enable the binary logging feature.

   Execute the following statement to view the format of binary logs:

   ```
   show variables like "binlog_format";
   ```

   Returned result:

   - *ROW*: The format of binary logs is row.
   - *STATEMENT*: The format of binary logs is statement.
   - *MIXED*: The format of binary logs is mixed.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add data sources.

4.6.4. Configure data sources for data synchronization from ApsaraDB for OceanBase

Before you use DataWorks to synchronize data from an ApsaraDB for OceanBase data source to AnalyticDB for MySQL V3.0, you need to configure data sources based on the operations in this topic. The configurations of data sources involve network environments, whitelists, and permissions.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

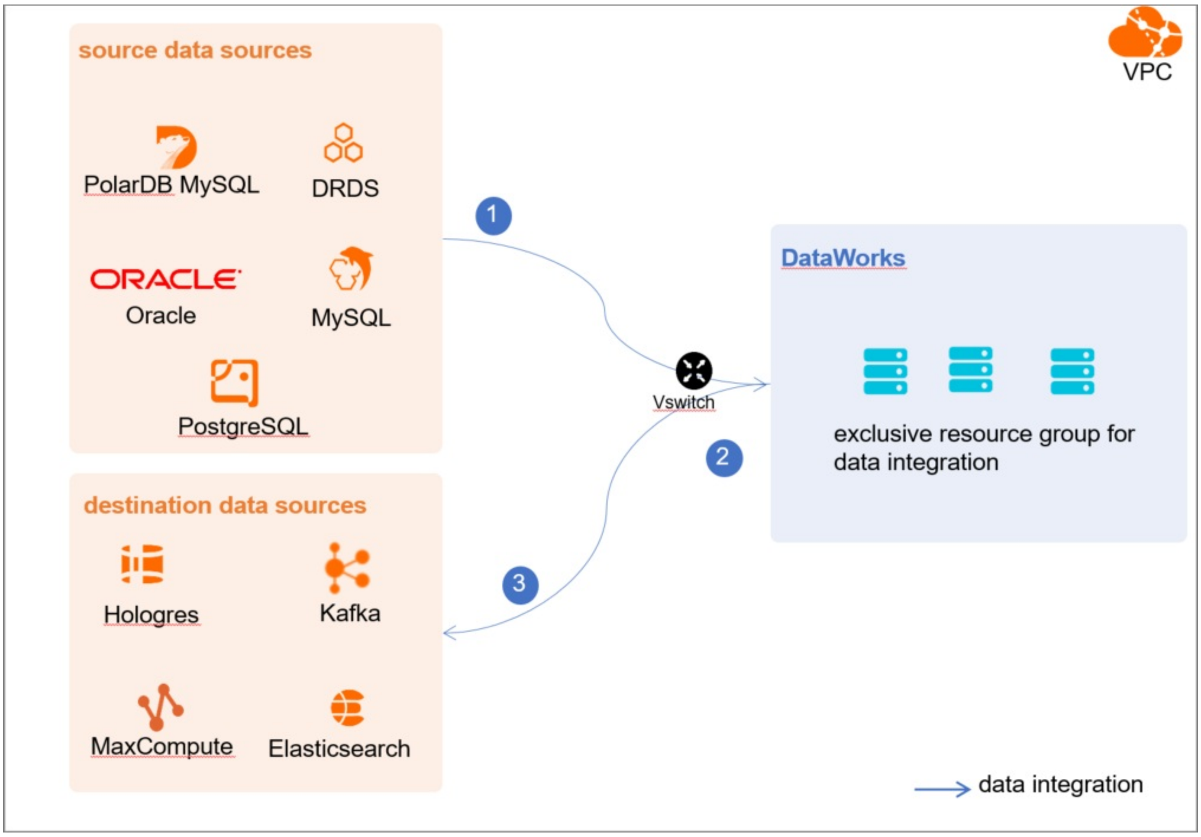
- Prepare data sources: A source ApsaraDB for OceanBase cluster and a destination AnalyticDB for MySQL V3.0 cluster are prepared.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and the exclusive resource group for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for the data sources

If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

Limits

ApsaraDB for OceanBase is a distributed relational database service that can integrate data distributed in multiple physical databases into a unified logical database. However, you can synchronize data of only one physical ApsaraDB for OceanBase database to an AnalyticDB for MySQL V3.0 cluster in real time.

Procedure

1. Configure a whitelist for the source ApsaraDB for OceanBase cluster.

   Perform the following steps to configure a whitelist:

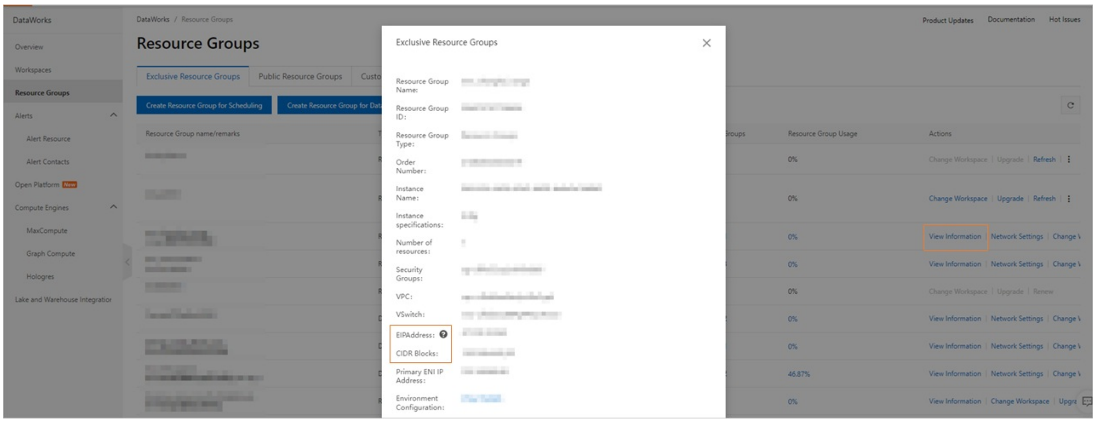
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

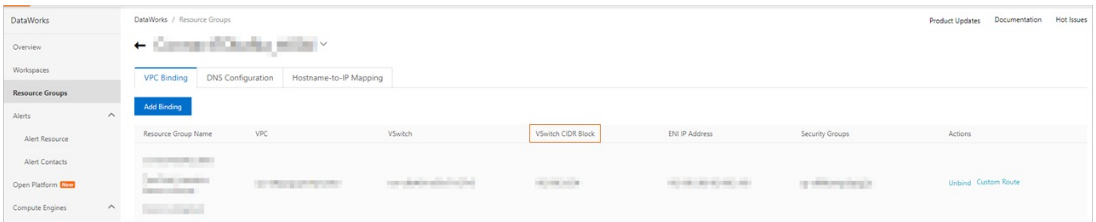
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration to the whitelist of the ApsaraDB for OceanBase cluster. For more information, see the "Set the whitelist" section of the Cluster workspace overview topic.

2. Create an account and grant the required permissions to the account.

   You must create an account to log on to the ApsaraDB for OceanBase cluster. You must also grant the required permissions to the account. For more information, see Create an account.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

4.6.5. Add data sources

Before you configure a real-time synchronization node to synchronize data from a data source to an AnalyticDB for MySQL V3.0 data source, you must add both data sources to DataWorks for subsequent source and destination configurations.

Prerequisites

Before you add data sources, make sure that the following operations are performed:

- Prepare data sources: A source data source and a destination data source are created.
- Create and grant permission to an account: An account that is used to access data sources is created.

Precautions

DataWorks provides workspaces in basic mode and standard mode. A workspace in basic mode does not isolate the development environment from the production environment. A workspace in standard mode isolates the development environment from the production environment.

If you use a workspace in standard mode, you must separately add data sources to the development environment and production environment.

Add a source PolarDB for MySQL data source

To add a source PolarDB for MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source PolarDB for MySQL data source, see Add a PolarDB data source.

If the source PolarDB for MySQL data source that you want to add fails the connectivity test, follow the instructions described in What can I do if a PolarDB data source cannot be connected to? to handle the exception.

Add a source MySQL data source

To add a source MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source MySQL data source, see Add a MySQL data source.

Add ApsaraDB for OceanBase as the source

To add ApsaraDB for OceanBase as the source, you must configure information such as the network connection type, and the access account and password as planned. For more information about how to add ApsaraDB for OceanBase as the source, see Add an ApsaraDB for OceanBase data source.

Add a destination AnalyticDB for MySQL V3.0 data source

For more information about how to add a destination AnalyticDB for MySQL V3.0 data source, see Configure an AnalyticDB for MySQL 3.0 connection.

What's next

After you add data sources, you can create and run a data sync node to synchronize data from the source data source to the destination data source.

For more information, see Create and configure a synchronization solution.

4.6.6. Create and configure a synchronization solution

After you configure data sources, network environments, and resource groups, you can create and run a synchronization solution. This topic describes how to create a synchronization solution and view the status of the nodes that are generated by the synchronization solution.

Prerequisites

Before you create a synchronization solution, make sure that the following operations are performed:

- Plan and configure resources
- Configure data sources for data synchronization from PolarDB
- Configure data sources for data synchronization from MySQL
- Configure data sources for data synchronization from ApsaraDB for OceanBase
- Add data sources

Create a synchronization solution

1.
2.
3. In the first step of the **Create Data Synchronization Solution** wizard, click **One-click realtime synchronization to AnalyticDB MySQL 3.0**.
4.

5. Select a data source as the source and configure synchronization rules.

   i. In the **Data Source** section, set the **Type** and **Data source** parameters.

   > Note You can set the Type parameter only to MySQL, ApsaraDB for OceanBase, or PolarDB.

   ii.

   iii.

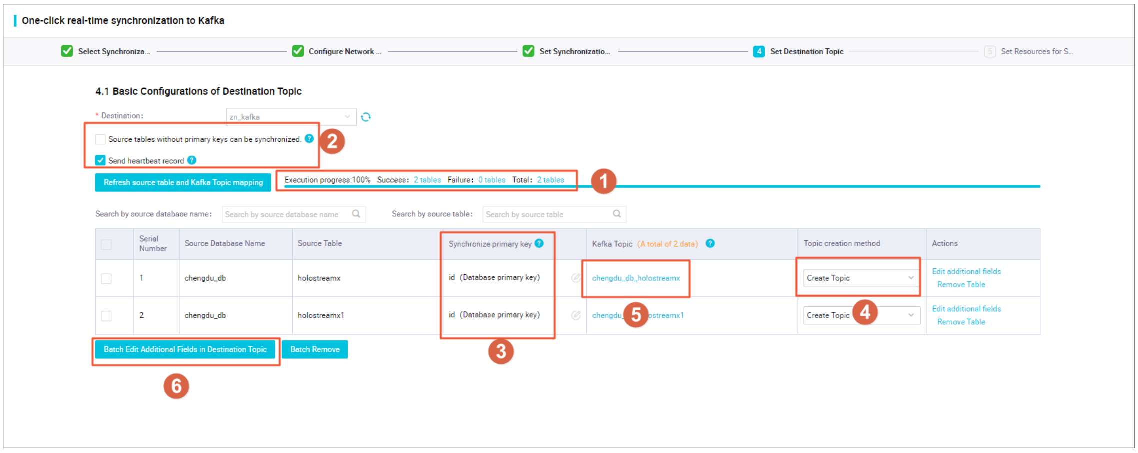
   iv.

6. Select a data source as the destination and configure the formats for the destination tables.

   i. In the **Set Destination Table** step, set the **Target AnalyticDB for MySQL 3.0 data source** parameter.

   ii. Click **Refresh source table and AnalyticDB MySQL 3.0 Table Mapping** to configure the mappings between the source tables and the destination AnalyticDB for MySQL V3.0 tables.

   iii. View the mapping progress, source tables, and mapped destination tables.



One-click real-time synchronization to Kafka

☑ Select Synchroniza... — ☑ Configure Network ... — ☑ Set Synchronizatio... — 4 Set Destination Topic — 5 Set Resources for S...

4.1 Basic Configurations of Destination Topic

* Destination: zn_kafka

□ Source tables without primary keys can be synchronized.

☑ Send heartbeat record

Refresh source table and Kafka Topic mapping

Execution progress:100% Success: 2 tables Failure: 0 tables Total: 2 tables

Search by source database name: Search by source database name

Search by source table: Search by source table

□	Serial Number	Source Database Name	Source Table	Synchronize primary key	Kafka Topic (A total of 2 data)	Topic creation method	Actions
□	1	chengdu_db	holostreamx	id (Database primary key)	chengdu_db_holostreamx	Create Topic	Edit additional fields Remove Table
□	2	chengdu_db	holostreamx1	id (Database primary key)	chengdu_ ostreamx1	Create Topic	Edit additional fields Remove Table

Batch Edit Additional Fields in Destination Topic

Batch Remove

Area No.	Description
1	The progress of mapping the source tables to the destination tables. Note The mapping may require an extended period of time if you want to synchronize data from a large number of tables.
②	■ If the tables in the source database contain primary keys, the system removes duplicate data based on the primary keys during the synchronization. ■ If the tables in the source database do not contain primary keys, you can click the ✎ icon to customize primary keys. You can use one field or a combination of several fields as the primary keys of the tables. This way, the system removes duplicate data based on the primary keys during the synchronization.
③	The method that is used to create a destination table. Valid values: ■ If you set the **Table creation method** parameter to **Use Existing Table**, the name of the automatically created AnalyticDB for MySQL V3.0 table is displayed in the **AnalyticDB for MySQL 3.0Table name** column. You can also select the table name that you want to use from the drop-down list. If you select **Use Existing Table**, the values of the **Table Type** and **Distribution Field Column** parameters cannot be changed. ■ If you set the **Table creation method** parameter to **Create Table**, the name of the automatically created AnalyticDB for MySQL V3.0 table is displayed. You can click the table name to view and modify the table creation statement. You can also set the **Table Type** and **Distribution Field Column** parameters for the table.

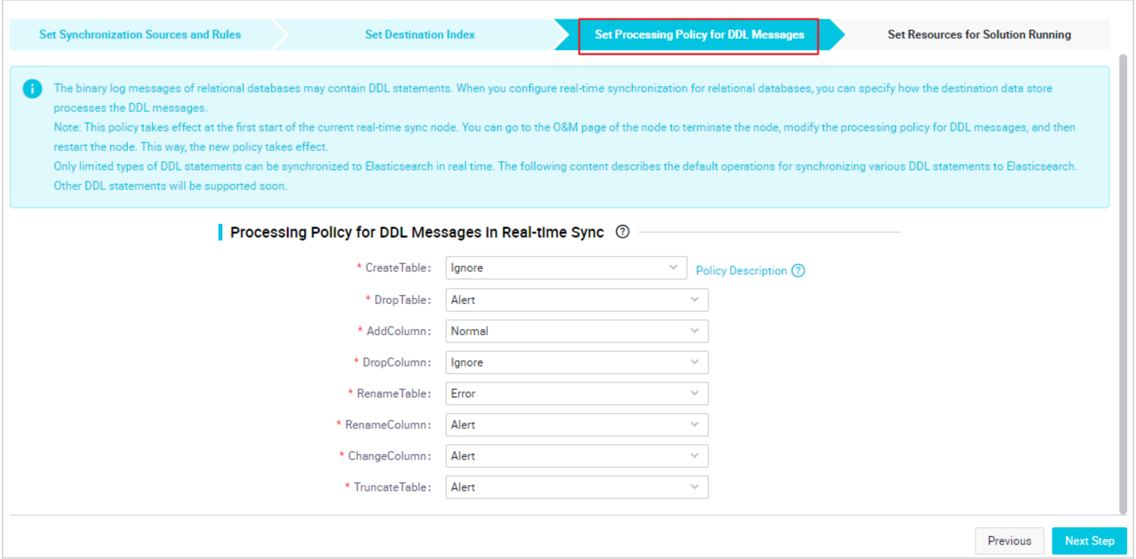
   iv. Click **Next Step**.

7. Configure rules for processing DDL messages.

   Sources, such as MySQL, PolarDB, and ApsaraDB for OceanBase, may contain multiple DDL messages. Before you synchronize data, you can configure different rules to process DDL messages based on your business requirements.

   > ⓘ **Note** The rules apply when a real-time synchronization solution is run for the first time. If you want to modify the rules in subsequent operations, go to the configuration page of the real-time synchronization solution to perform the operation. For more information, see Manage the synchronization solution.

   i. In the **Set Processing Policy for DDL Messages** step, configure rules to process DDL messages during data synchronization.


Set Synchronization Sources and Rules
Set Destination Index
Set Processing Policy for DDL Messages
Set Resources for Solution Running

The binary log messages of relational databases may contain DDL statements. When you configure real-time synchronization for relational databases, you can specify how the destination data store processes the DDL messages.
Note: This policy takes effect at the first start of the current real-time sync node. You can go to the O&M page of the node to terminate the node, modify the processing policy for DDL messages, and then restart the node. This way, the new policy takes effect.
Only limited types of DDL statements can be synchronized to Elasticsearch in real time. The following content describes the default operations for synchronizing various DDL statements to Elasticsearch. Other DDL statements will be supported soon.

Processing Policy for DDL Messages In Real-time Sync

* CreateTable: Ignore
* DropTable: Alert
* AddColumn: Normal
* DropColumn: Ignore
* RenameTable: Error
* RenameColumn: Alert
* ChangeColumn: Alert
* TruncateTable: Alert

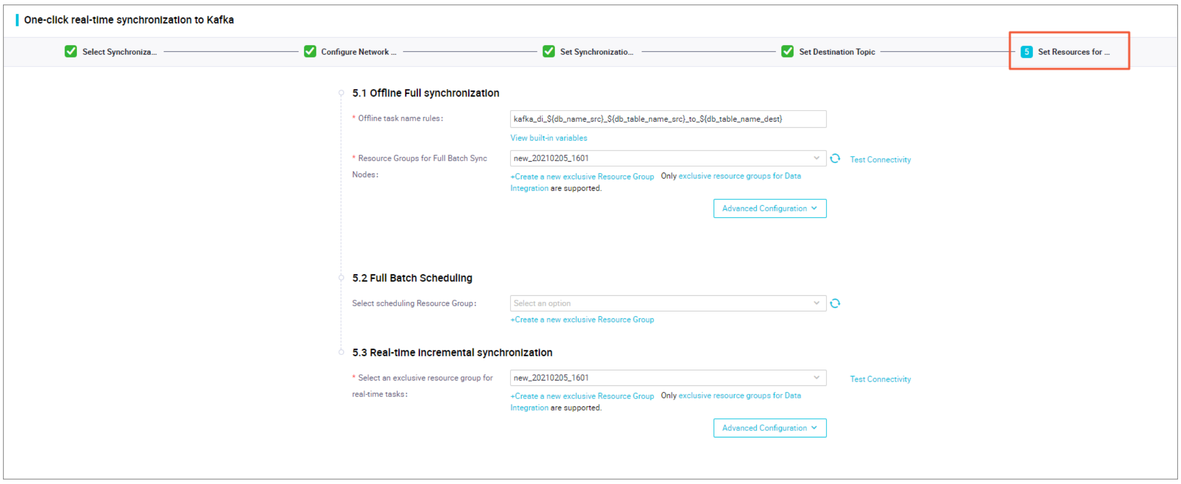
The following table describes the processing rules for different DDL messages.

DDL message	Rule
CreateTable	DataWorks processes a DDL message of the related type based on the following rules after it receives the message: ▪ **Normal**: sends the message to the destination. Then, the destination processes the message. Each destination may process DDL messages based on its own business logic. If you select Normal for CreateTable, DataWorks only forwards the messages. ▪ **Ignore**: ignores the message and does not send it to the destination. ▪ **Alert**: ignores the message and records the alert in real-time synchronization logs. In addition, the alert contains information about the reason indicating that a message is ignored because of a running error. ▪ **Error**: returns an error when the real-time sync solution is running and terminates the real-time sync solution.
DropTable	
AddColumn	
DropColumn	
RenameTable	
RenameColumn	
ChangeColumn	
TruncateTable	

   ii. Click **Next Step**.

8. Configure the resources required by the synchronization solution.

   In the **Set Resources for Solution Running** step, set the parameters that are described in the following tables.



One-click real-time synchronization to Kafka

5.1 Offline Full synchronization

* Offline task name rules: kafka_di_${db_name_src}_${db_table_name_src}_to_${db_table_name_dest}

* Resource Groups for Full Batch Sync Nodes: new_20210205_1601

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

5.2 Full Batch Scheduling

Select scheduling Resource Group: Select an option

+Create a new exclusive Resource Group

5.3 Real-time Incremental synchronization

* Select an exclusive resource group for real-time tasks: new_20210205_1601

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

- Offline Sync

Parameter	Description
Offline task name rules	The name of the batch sync node that is used to synchronize the full data of the source. After a sync solution is created, DataWorks first generates a batch sync node to synchronize full data, and then generates real-time sync nodes to synchronize incremental data.
Resource Groups for Full Batch Sync Nodes	The exclusive resource group for Data Integration that is used to run the batch sync node. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Scheduling Settings

Parameter	Description
Select scheduling Resource Group	The resource group for scheduling that is used to run the nodes. Only exclusive resource groups for Data Integration can be used to run sync solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Incremental Sync

Parameter	Description
Resource Groups for Incremental Batch Sync Nodes	The exclusive resource group that is used to run the real-time sync nodes. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Channel Settings

Operation	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: **20**.
Number of concurrent writes on the target side	The maximum number of JDBC connections that are allowed for the destination. The number must be larger than or equal to the number specified for the source. Specify an appropriate number based on the resources of the destination. Default value: **45**.

9. Click **Complete Configuration**. The real-time sync solution used to synchronize all data in a database is created.

Run the synchronization solution

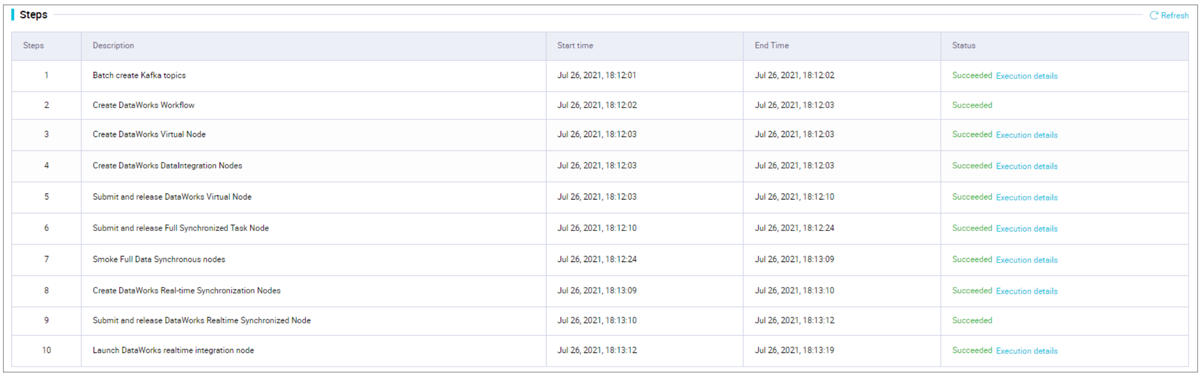
On the **Solution task list** tab of the Tasks page, find the created synchronization solution and click **Submit and Run** in the Operation column to run the synchronization solution.

If the execution of the solution fails, you can view the error message and troubleshoot the issues based on the following answers to some commonly asked questions:

- The system displays the following error message for a real-time synchronization node: com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX. What can I do?
- The system displays the following error message for a real-time synchronization node: com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation. What can I do?
- The system displays the following error message for a real-time synchronization node: com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first. What can I do?
- The system displays the following error message for a batch synchronization node: com.alibaba.datax.common.exception.DataXException: Code:[HoloWriter-02], Description:[Invalid config parameter in your configuration.]. - Field _log_file_name_offset_ not allow null but not present in user configured columns. What can I do?

View the status and results of the synchronization nodes

- On the **Tasks** page, find the solution that is run and click **Execution details** in the Operation column. Then, you can view the execution details of all nodes.



Steps

Steps	Description	Start time	End Time	Status
1	Batch create Kafka topics	Jul 26, 2021, 18:12:01	Jul 26, 2021, 18:12:02	Succeeded Execution details
2	Create DataWorks Workflow	Jul 26, 2021, 18:12:02	Jul 26, 2021, 18:12:03	Succeeded
3	Create DataWorks Virtual Node	Jul 26, 2021, 18:12:03	Jul 26, 2021, 18:12:03	Succeeded Execution details
4	Create DataWorks DataIntegration Nodes	Jul 26, 2021, 18:12:03	Jul 26, 2021, 18:12:03	Succeeded Execution details
5	Submit and release DataWorks Virtual Node	Jul 26, 2021, 18:12:03	Jul 26, 2021, 18:12:10	Succeeded Execution details
6	Submit and release Full Synchronized Task Node	Jul 26, 2021, 18:12:10	Jul 26, 2021, 18:12:24	Succeeded Execution details
7	Smoke Full Data Synchronous nodes	Jul 26, 2021, 18:12:24	Jul 26, 2021, 18:13:09	Succeeded Execution details
8	Create DataWorks Real-time Synchronization Nodes	Jul 26, 2021, 18:13:09	Jul 26, 2021, 18:13:10	Succeeded Execution details
9	Submit and release DataWorks Realtime Synchronized Node	Jul 26, 2021, 18:13:10	Jul 26, 2021, 18:13:12	Succeeded
10	Launch DataWorks realtime integration node	Jul 26, 2021, 18:13:12	Jul 26, 2021, 18:13:19	Succeeded Execution details

- Find a node whose execution details you want to view and click **Execution details** in the Status column. In the message that appears, click the provided link to go to the DataStudio page.

Manage the synchronization solution

- View or edit the synchronization solution.

  On the **Solution task list** tab of the Tasks page, find the created synchronization solution and choose **More > View Configuration** or choose **More > Modify Configuration** in the Operation column. Then, you can view or modify the configurations of the synchronization solution.

> Note You can click **Modify Configuration** to modify a synchronization node only if the synchronization node is in the **Not Running** state. If you click Modify Configuration in the Operation column of a synchronization node in another state, you can view only the information about that synchronization node.

- Delete the synchronization solution.

  Find the real-time synchronization solution that you want to delete and choose **More** > **Delete** in the Operation column. In the **Delete** message, click **OK**.

  > Note After you click OK, only the configuration record of the synchronization solution is deleted. The generated synchronization nodes and tables are not affected.

- Change the priority for the batch synchronization solution

  Find the newly created batch synchronization solution and choose **More** > **Change Priority** in the Operation column. In the **Change Priority** dialog box, enter the desired priority and click **Confirm**. You can set the priority to an integer from 1 to 8. A larger value indicates a higher priority.

  > Note If multiple batch synchronization solutions have the same priority, the system runs them based on the order they are committed.

4.6.7. FAQ

This topic provides answers to frequently asked questions (FAQ) about data synchronization to AnalyticDB for MySQL V3.0 data sources.

- What can I do if a PolarDB data source cannot be connected to?
- What can I do if an ApsaraDB for OceanBase data source cannot be connected to?
- What can I do if a MySQL data source cannot be connected to?
- The system displays the following error message for a real-time synchronization node: com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX. What can I do?
- The system displays the following error message for a real-time synchronization node: com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation. What can I do?
- The system displays the following error message for a real-time synchronization node: com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first. What can I do?
- The system displays the following error message for a batch synchronization node: com.alibaba.datax.common.exception.DataXException: Code:[HoloWriter-02], Description:[Invalid config parameter in your configuration.]. - Field _log_file_name_offset_ not allow null but not present in user configured columns. What can I do?

What can I do if a PolarDB data source cannot be connected to?

- Problem description: The connectivity test fails when you add a PolarDB data source.
- Solution: Switch to the connection string mode and check the whitelist configuration of the PolarDB data source and the virtual private cloud (VPC) configuration of your exclusive resource group.

What can I do if an ApsaraDB for OceanBase data source cannot be connected to?

- Problem description: The connectivity test fails when you add an ApsaraDB for OceanBase data source.
- Solution: Switch to the connection string mode and check the whitelist configuration of the PolarDB data source and the virtual private cloud (VPC) configuration of your exclusive resource group.

What can I do if a MySQL data source cannot be connected to?

- Problem description: The connectivity test fails when you add a MySQL data source.
- Solution: Switch to the connection string mode and check the whitelist configuration of the PolarDB data source and the virtual private cloud (VPC) configuration of your exclusive resource group.

The system displays the following error message for a real-time synchronization node: com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX. What can I do?

- Problem description: The real-time synchronization node fails to run, and the system displays the error message `com.alibaba.otter.canal.parse.exception.PositionNotFoundException: can't find start position for XXX` .
- Cause: The binary logging feature is disabled for the PolarDB data source.
- Solution: Enable the binary logging feature for the PolarDB data source. For more information, see Configure data sources for data synchronization from PolarDB. In addition, you must change one or more data records and change the start time for running the real-time synchronization node to the current time.

The system displays the following error message for a real-time synchronization node: com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation. What can I do?

- Problem description: The real-time synchronization node fails to run, and the system displays the error message `com.alibaba.otter.canal.parse.exception.CanalParseException: command : 'show master status' has an error! pls check. you need (at least one of) the SUPER,REPLICATION CLIENT privilege(s) for this operation` .
- Cause: The account used to synchronize data is not authorized to access the PolarDB data source, or the connected PolarDB database is not deployed on the PolarDB Writer node.
- Solution: Authorize the account to access the PolarDB data source. For more information, see Configure data sources for data synchronization from PolarDB. You can also check whether the connected PolarDB database is deployed on the PolarDB Writer node. When a real-time synchronization node is running, the system cannot capture data from the PolarDB Reader nodes of the PolarDB data source.

The system displays the following error message for a real-time synchronization node: com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first. What can I do?

- Problem description: The real-time synchronization node fails to run, and the system displays the error message `com.alibaba.datax.plugin.reader.mysqlbinlogreader.MysqlBinlogReaderException: The mysql server does not enable the binlog write function. Please enable the mysql binlog write function first` .
- Cause: The loose_polar_log_bin parameter is not specified for the PolarDB data source.
- Solution: Specify the loose_polar_log_bin parameter. For more information, see Configure data sources for data synchronization from PolarDB.

The system displays the following error message for a batch synchronization node: com.alibaba.datax.common.exception.DataXException: Code:[HoloWriter-02], Description:[Invalid config parameter in your configuration.]. - Field _log_file_name_offset_ not allow null but not present in user configured columns. What can I do?

- Problem description: The batch synchronization node fails to run, and the system displays the error message `com.alibaba.datax.common.exception.DataXException: Code:[HoloWriter-02], Description:[Invalid config parameter in your configuration.]. - Field _log_file_name_offset_ not allow null but not present in user configured columns` .
- Cause: The engine-type plug-in of DataWorks for batch synchronization is not upgraded to the latest version.
- Solution: submit a ticket to upgrade the plug-in.

4.7. Synchronize data to MaxCompute

4.7.1. Preparations

4.7.1.1. Plan and configure resources

When you use the sync solutions of DataWorks to synchronize data, you can use only exclusive resource groups for Data Integration to run data integration nodes. In addition, you can select a shared or exclusive resource group for scheduling based on your business requirements. This topic describes the resources that are used for sync solutions and how to configure the resources.

Context

- Resource planning and preparation

  When you use sync solutions to synchronize data, data integration nodes run on resources in resource groups for Data Integration and resource groups for scheduling. In this case, you can use only exclusive resource groups for Data Integration. Before you synchronize data, you must purchase exclusive resources for Data Integration and create an exclusive resource group for Data Integration in your DataWorks workspace.

  For more information about exclusive resource groups for Data Integration, see Exclusive resources for Data Integration.

- Network connections

  An exclusive resource group for Data Integration is essentially a group of Elastic Compute Service (ECS) instances. After you purchase and create such an exclusive resource group, it is isolated from other services. You must associate the resource group with a virtual private cloud (VPC) to ensure network connectivity between the resource group and data sources during subsequent data synchronization.

What's next

After you plan and configure resources, you can configure data sources. You must connect the exclusive resource group for Data Integration to the source and destination. You must also create an account and grant the required permissions to the account. This account is used to access the source and destination. The preceding operations help create sync nodes. You can synchronize data only from PolarDB, Oracle, or MySQL to MaxCompute. You can select a data source based on your business requirements. For more information about how to configure a data source, see Configure a data source (PolarDB), Configure a source Oracle data source, or Configure data sources for data synchronization from MySQL.

4.7.1.2. Configure a source PolarDB data source

If you synchronize data in PolarDB to MaxCompute, the source data source is PolarDB, and the destination data source is MaxCompute. Before you run a data synchronization node, you must refer to the operations in this topic to prepare the configurations such as network environments and whitelists for data sources.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

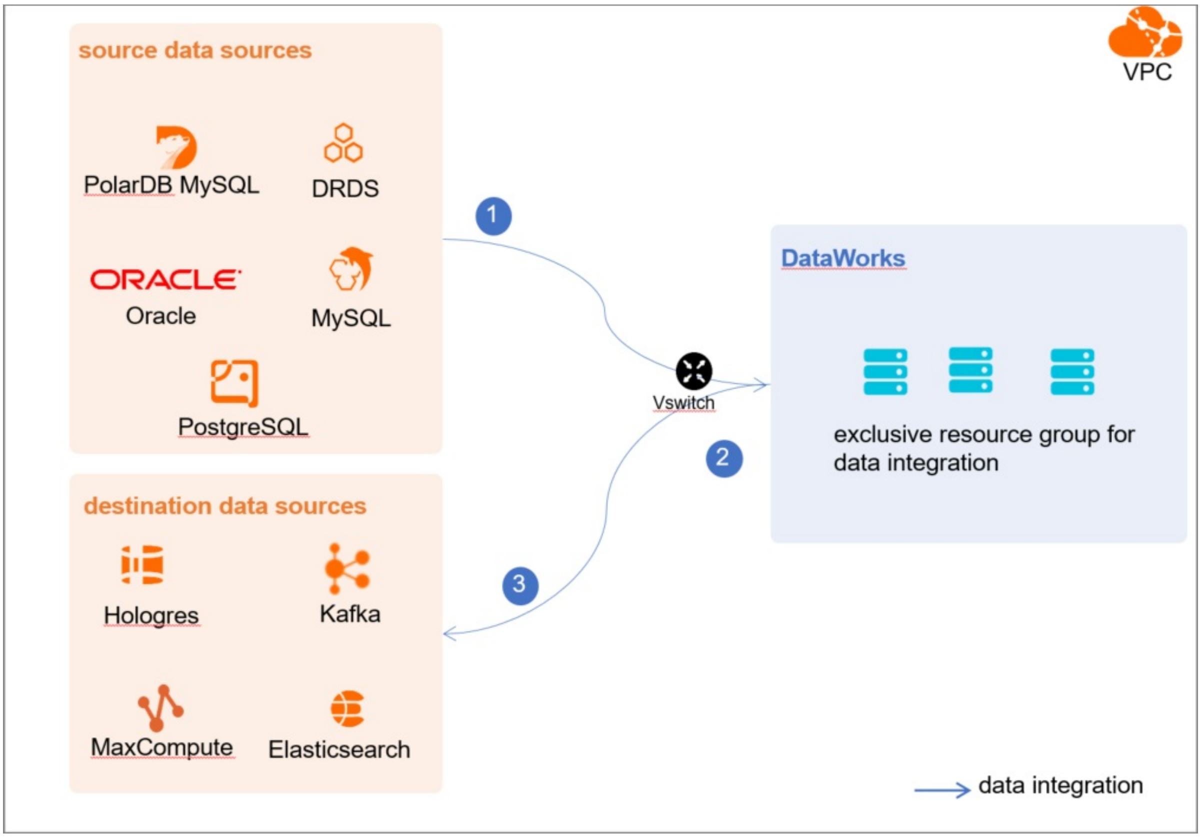
- Prepare data sources: A PolarDB for MySQL cluster and a MaxCompute project are created. In this topic, a PolarDB for MySQL cluster is used as the source data source.
- Plan and prepare resources: An exclusive resource group for data integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, connect data sources to exclusive resource groups for data integration based on your business requirements. After data sources and exclusive resource groups for data integration are connected, you can refer to the operations in this topic to configure access settings such as vSwitches and whitelists.
  - If data sources and exclusive resource groups for data integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If data sources and exclusive resource groups for data integration reside in different network environments, you must connect data sources and resource groups by using methods such as a VPN gateway.
- Prepare the MaxCompute client: The MaxCompute client is installed. You need to use the MaxCompute client to configure attributes for the destination MaxCompute data source. For more information, see MaxCompute client.

Context

Before you synchronize data from source data sources to destination data sources, make sure that data sources and exclusive resource groups are connected. In addition, you must make sure that exclusive resource groups can be used to access data sources.

- Configure whitelists for data sources

  If data sources and exclusive resource groups for data integration reside in the same VPC, you need to add the CIDR block of the exclusive resource group to the whitelists of data sources. This ensures that the exclusive resource group for data integration can be used to access data sources.



- Create an account and authorize the account

  You must create an account that can be used to access data sources, read data from the source data source, and write data to the destination data source in the data integration process.

- Enable the binary logging feature

  If the source data source is a PolarDB for MySQL cluster, you must enable the binary logging feature for the cluster. Alibaba Cloud PolarDB for MySQL is fully compatible with MySQL and uses high-level physical logs to replace binary logs. To facilitate the integration between PolarDB and the MySQL ecosystem, you can enable the binary logging feature for PolarDB clusters.

Limits

- Only PolarDB for MySQL clusters can be used as source data sources. In this topic, PolarDB indicates PolarDB for MySQL data sources.
- Only data stored on the primary node of the PolarDB cluster can be synchronized.

Configure the source PolarDB data source

1. Configure a whitelist for the PolarDB for MySQL cluster.

   To add the CIDR block of the VPC where the exclusive resource group for Data Integration resides to a whitelist of the PolarDB for MySQL cluster, perform the following steps:

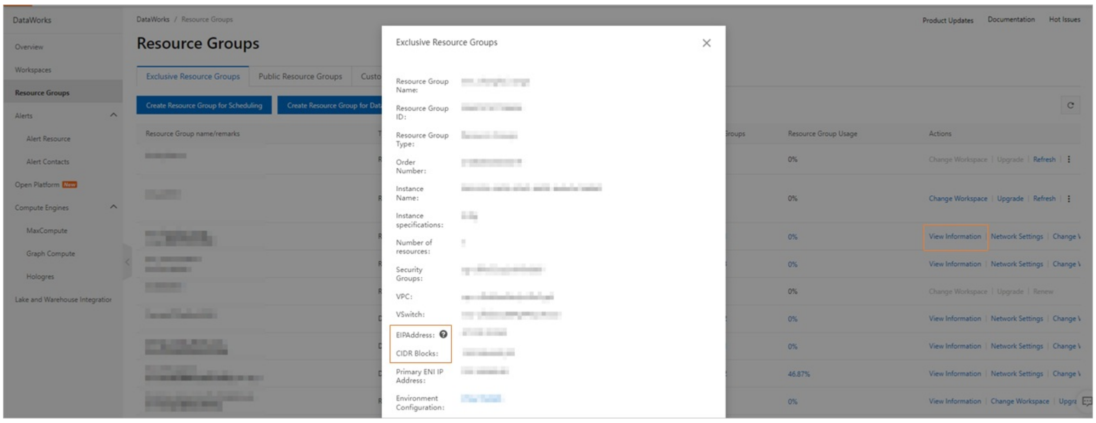
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

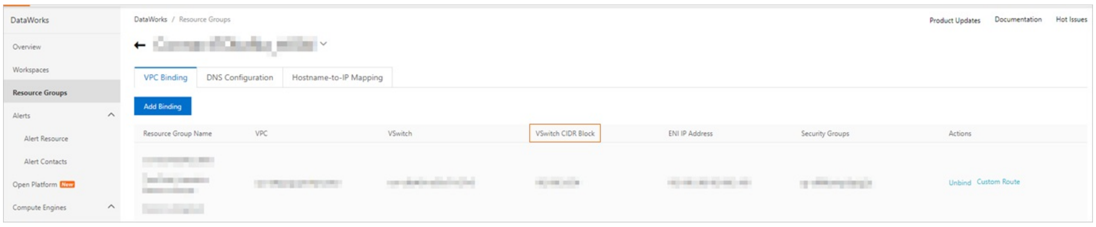
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.

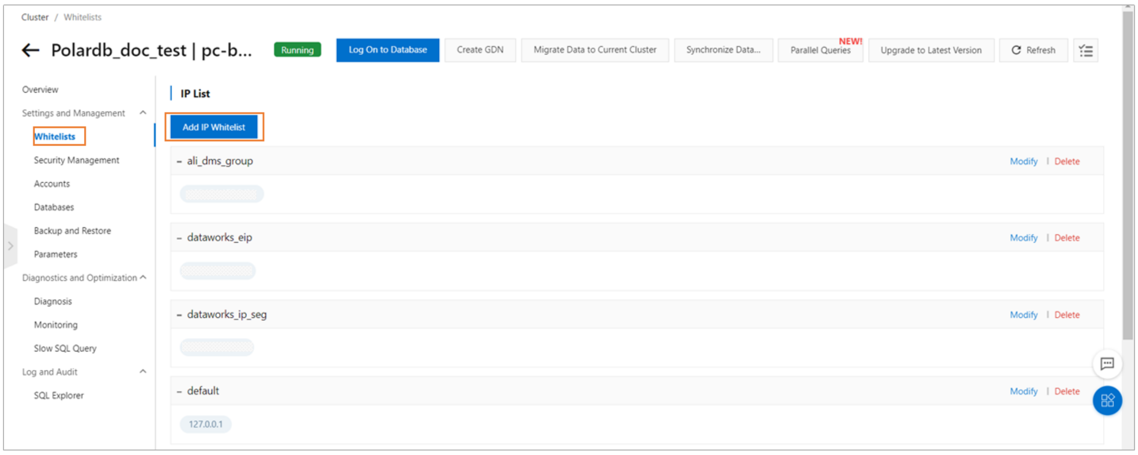


e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR blocks recorded in the preceding steps to the whitelist of the PolarDB for MySQL cluster.



For more information, see Configure an IP whitelist.

2. Create a PolarDB for MySQL database account.

   For more information, see Create a database account.

3. Enable the binary logging feature for the PolarDB cluster.

   For more information, see Enable binary logging.

Configure the destination MaxCompute data source

1. Log on to the MaxCompute client by using the account of a project owner.

   For more information, see MaxCompute client.

2. Enable the atomicity, consistency, isolation, durability (ACID) property for the MaxCompute project.

   Run the following command on the MaxCompute client:

   ```
   setproject odps.sql.acid.table.enable=true;
   ```

3. (Optional)Enable the MaxCompute V2.0 data type edition.

   If you need to use the TIMESTAMP data type in MaxCompute V2.0, run the following command to enable the MaxCompute V2.0 data type edition:

   ```
   setproject odps.sql.type.system.odps2=true;
   ```

4. Create an Alibaba Cloud account.

   This account is used to add a data source and access MaxCompute for data synchronization. For more information about how to create an Alibaba Cloud account, see Create an Alibaba Cloud account.

   After the Alibaba Cloud account is created, you can record the AccessKey ID and AccessKey secret of the account for future use.

What's next

After data sources are configured, the source data source, destination data source, and exclusive resource group for data integration are connected. Then, the exclusive resource group for data integration can be used to access data sources. You can add the source data source and destination data source to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

4.7.1.3. Configure data sources for data synchronization from Oracle

Before you synchronize data from Oracle to MaxCompute, you can refer to the operations described in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions. You must configure a source Oracle data source and a destination MaxCompute data source.

Prerequisites

Before you configure data sources, make sure that the following operations are performed:

- Prepare data sources: An Oracle database and a MaxCompute project are prepared.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.
- Prepare the MaxCompute client: The MaxCompute client is installed. You need to use the MaxCompute client to configure attributes for the destination MaxCompute data source. For more information, see MaxCompute client.

Context

Configure a source Oracle data source

Configure the destination MaxCompute data source

1. Log on to the MaxCompute client by using the account of a project owner.

For more information, see MaxCompute client.

2. Enable the atomicity, consistency, isolation, durability (ACID) property for the MaxCompute project.

   Run the following command on the MaxCompute client:

   ```
   setproject odps.sql.acid.table.enable=true;
   ```

3. (Optional)Enable the MaxCompute V2.0 data type edition.

   If you need to use the TIMESTAMP data type in MaxCompute V2.0, run the following command to enable the MaxCompute V2.0 data type edition:

   ```
   setproject odps.sql.type.system.odps2=true;
   ```

4. Create an Alibaba Cloud account.

   This account is used to add a data source and access MaxCompute for data synchronization. For more information about how to create an Alibaba Cloud account, see Create an Alibaba Cloud account.

   After the Alibaba Cloud account is created, you can record the AccessKey ID and AccessKey secret of the account for future use.

What's next

After data sources are configured, the source data source, destination data source, and exclusive resource group for data integration are connected. Then, the exclusive resource group for data integration can be used to access data sources. You can add the source data source and destination data source to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

4.7.1.4. Configure data sources for data synchronization from MySQL

Before you synchronize data from a MySQL data source to MaxCompute, you can refer to the operations in this topic to configure data sources. The configurations of data sources include network environments, whitelists, and permissions.

Prerequisites

Before you configure the data sources, make sure that the following operations are performed:

- Prepare data sources: A MySQL data source and a destination MaxCompute data source are created.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
    - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
    - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.
- Execute the following statement to check the version of your MySQL database. You can use a `MySQL 5.X` or `MySQL 8.X` database.

  ```
  select version();
  ```

  > **Note** Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.X` or `V8.X`. PolarDB for MySQL is not supported. If the MySQL version of your ApsaraDB RDS for MySQL database is not `V5.X` or `V8.X`, use an ApsaraDB RDS for MySQL database whose MySQL version is `V5.X` or `V8.X`. Otherwise, the data synchronization node fails to run.

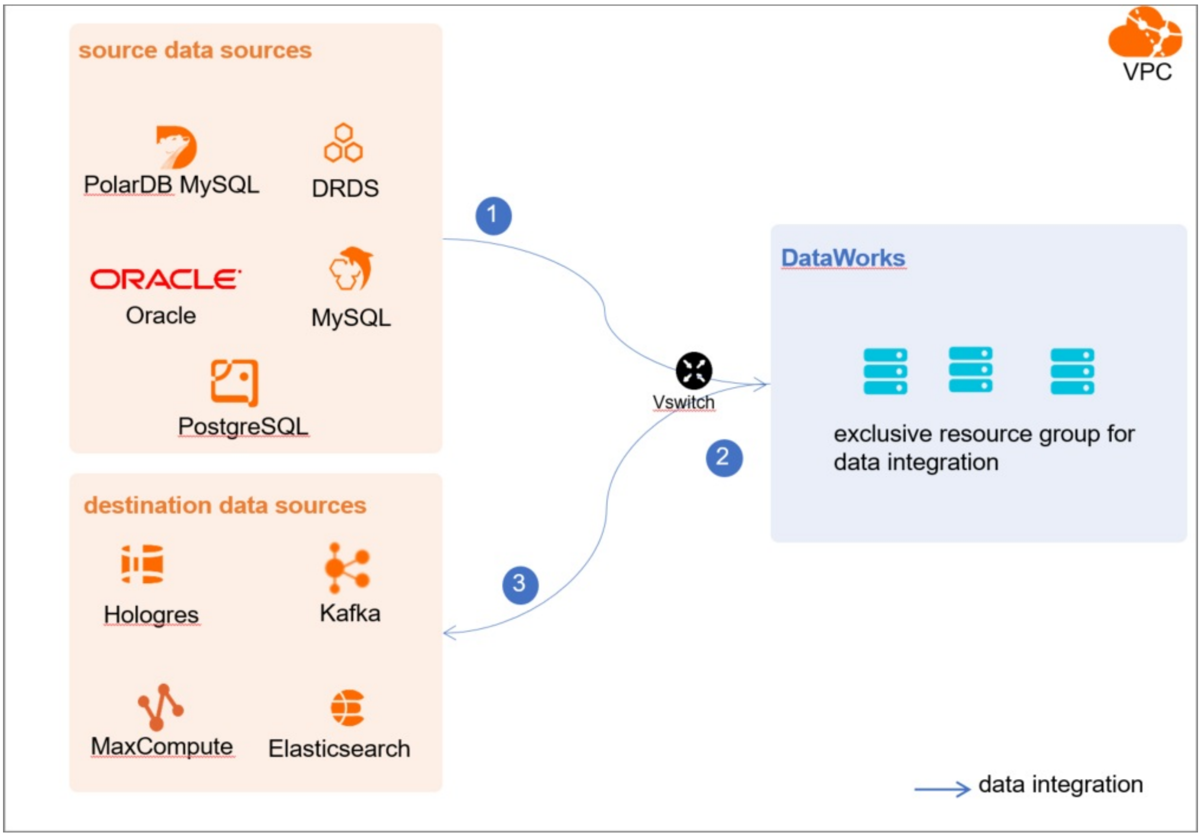
- Prepare the MaxCompute client: The MaxCompute client is installed. You need to use the MaxCompute client to configure attributes for the destination MaxCompute data source. For more information, see MaxCompute client.

Context

Before you synchronize data from source data sources to destination data sources, make sure that data sources and exclusive resource groups for data integration are connected. You must also make sure that the exclusive resource groups for data integration can be used to access the data sources.

- Configure whitelists for the data sources

  If the data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source data source is a MySQL database, you must enable the binary logging feature. Binary logs record changes to all table schemas and modifications on table data. You can execute statements such as CREATE and ALTER to perform operations on table schemas. You can execute statements such as INSERT, UPDATE, and DELETE to perform operations on table data. You can use binary logs to view the change history of the database, back up incremental data and restore data in the database, and replicate data from the primary database to secondary databases.

  Formats of binary logs:

  - Statement: SQL statement-based replication. Binary logs in this format record the SQL statements that are executed to modify data entries.
  - Row: row-based replication. Binary logs in this format record only modification details about data entries in rows.
  - Mixed: replication in mixed mode. This mode combines the statement and row formats. In most cases, binary logs in the statement format are used to record the SQL statements that are executed to modify data entries, such as functions. If data replication from the primary database to secondary databases cannot be implemented by using binary logs in this format, switch to the row format. MySQL determines which format to use based on each SQL statement that is executed.

Limits

Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.x` or `V8.x` . PolarDB for MySQL is not supported.

Configure a source MySQL data source

1. Configure a whitelist for the MySQL database.

   Add the CIDR block of the VPC in which the exclusive resource group resides to the whitelist of the MySQL database.

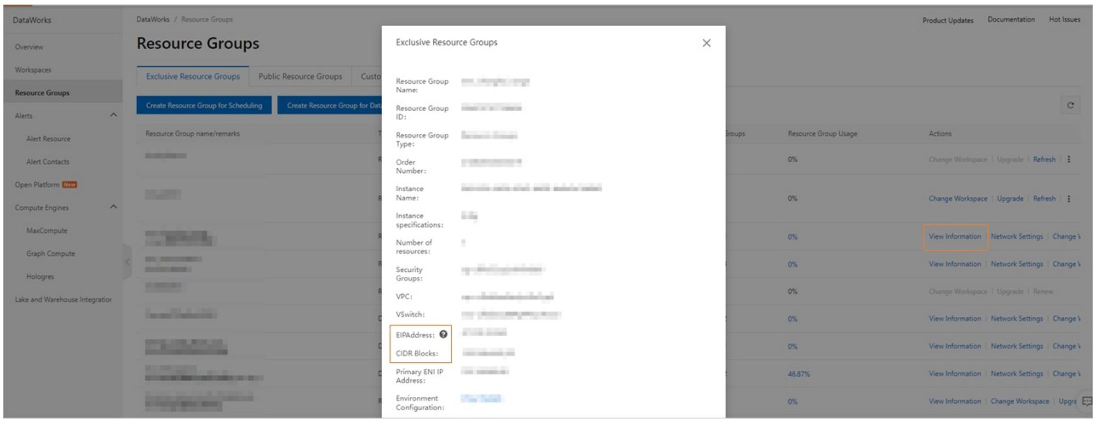
   i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

      a. Log on to the DataWorks console.

      b. In the left-side navigation pane, click **Resource Groups**.

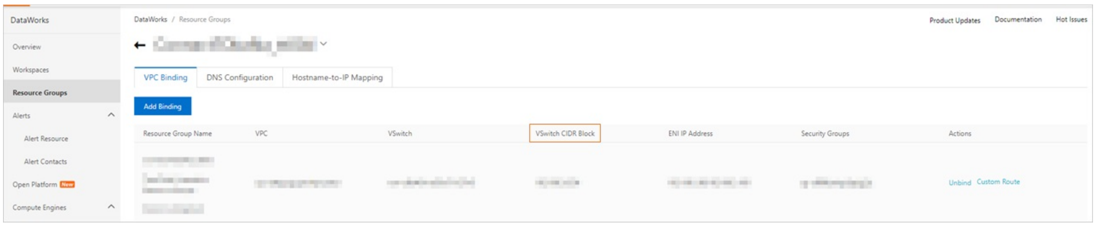
      c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

      d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



      e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

      f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



   ii. Add the EIP and CIDR block of the exclusive resource group to the whitelist of the MySQL database.

2. Create an account and grant permissions to the account.

   You must create an account to log on to the MySQL database. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

   i. Create an account.

      For more information, see Create an account to access a MySQL database.

ii. Grant permissions to the account.

You can run the following command to grant permissions to the account. Alternatively, you can grant the `SUPER` permission to the account. Replace `Account for data synchronization` with the created account when you execute the specific statement.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Password'; // Create an account that is used to synchronize data and set a password so that you can use the account and password to access the database from a host. % indicates a host.
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%'; // Grant the SELECT, REPLICATION SLAVE, and REPLICATION CLIENT permissions to the account.
```

`*.*` indicates that the synchronization account is granted the preceding permissions on all tables in all databases. You can also grant the preceding permissions on specified tables in the specified database to the synchronization account. For example, to grant the account the preceding permissions on the *user* table in the *test* database, execute the following statement: `GRANT SELECT, REPLICATION CLIENT ON test.user TO 'Account for data synchronization'@'%';`.

> **Note** The `REPLICATION SLAVE` permission is a global permission. You cannot grant this permission on specified tables in the specified database to the synchronization account.

3. Enable the binary logging feature for the MySQL database.

Perform the following steps to check whether the binary logging feature is enabled and to query the format of binary logs:

- Execute the following statement to check whether the binary logging feature is enabled:

```
show variables like "log_bin";
```

If *ON* is displayed in the returned result, the binary logging feature is enabled.

- If you use a secondary database to synchronize data, execute the following statement to check whether the binary logging feature is enabled:

```
show variables like "log_slave_updates";
```

If *ON* is displayed in the returned result, the binary logging feature is enabled for the secondary database.

If the returned result is different from the preceding result, follow the instructions described in the *MySQL documentation* to enable the binary logging feature.

Execute the following statement to view the format of binary logs:

```
show variables like "binlog_format";
```

Returned result:

- *ROW*: The format of binary logs is row.
- *STATEMENT*: The format of binary logs is statement.
- *MIXED*: The format of binary logs is mixed.

Configure the destination MaxCompute data source

1. Log on to the MaxCompute client by using the account of a project owner.

For more information, see MaxCompute client.

2. Enable the atomicity, consistency, isolation, durability (ACID) property for the MaxCompute project.

Run the following command on the MaxCompute client:

```
setproject odps.sql.acid.table.enable=true;
```

3. (Optional)Enable the MaxCompute V2.0 data type edition.

If you need to use the TIMESTAMP data type in MaxCompute V2.0, run the following command to enable the MaxCompute V2.0 data type edition:

```
setproject odps.sql.type.system.odps2=true;
```

4. Create an Alibaba Cloud account.

   This account is used to add a data source and access MaxCompute for data synchronization. For more information about how to create an Alibaba Cloud account, see Create an Alibaba Cloud account.

   After the Alibaba Cloud account is created, you can record the AccessKey ID and AccessKey secret of the account for future use.

What's next

After data sources are configured, the source data source, destination data source, and exclusive resource group for data integration are connected. Then, the exclusive resource group for data integration can be used to access data sources. You can add the source data source and destination data source to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add a data source, see Add a data source.

4.7.1.5. Add a data source

Before you configure a data synchronization solution to synchronize data from a source data source to a destination MaxCompute data source, you must add the source data source and the destination data source in the DataWorks console.

Prerequisites

Add a source Oracle data source

To add a source Oracle data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source Oracle data source, see Add an Oracle data source.

Add a source MySQL data source

To add a source MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source MySQL data source, see Add a MySQL data source.

Add a destination MaxCompute data source

For more information, see Add a MaxCompute data source.

What's next

After you add data sources, you can create and run a data sync node to synchronize data from the source data source to the destination data source.

For more information, see Synchronize data to MaxCompute in real time.

4.7.2. Synchronize data to MaxCompute in real time

After you configure data sources, network environments, and resource groups, you can create and run synchronization nodes. This topic describes how to configure a synchronization solution to synchronize data to MaxCompute in real time and view the status of the nodes generated by the synchronization solution.

Prerequisites

Before you create a synchronization solution, make sure that the following operations are performed:

- Plan and configure resources
- Configure a source PolarDB data source
- Configure data sources for data synchronization from Oracle
- Configure data sources for data synchronization from MySQL
- Add a data source

Limits

You can use only a self-manged MaxCompute data source that resides in the same region as your workspace. If you use a self-managed MaxCompute data source that resides in a different region from your workspace, the data source can be connected to the resource group that you use. However, an error indicating that the compute engine instance does not exist will be reported when the system creates a MaxCompute table during the running of the synchronization solution.

> ⓘ **Note** If you use a self-managed MaxCompute data source, you must associate a MaxCompute compute engine instance with your DataWorks workspace. Otherwise, an ODPS SQL node cannot be created. As a result, a node that is used to mark the end of full synchronization cannot be created.

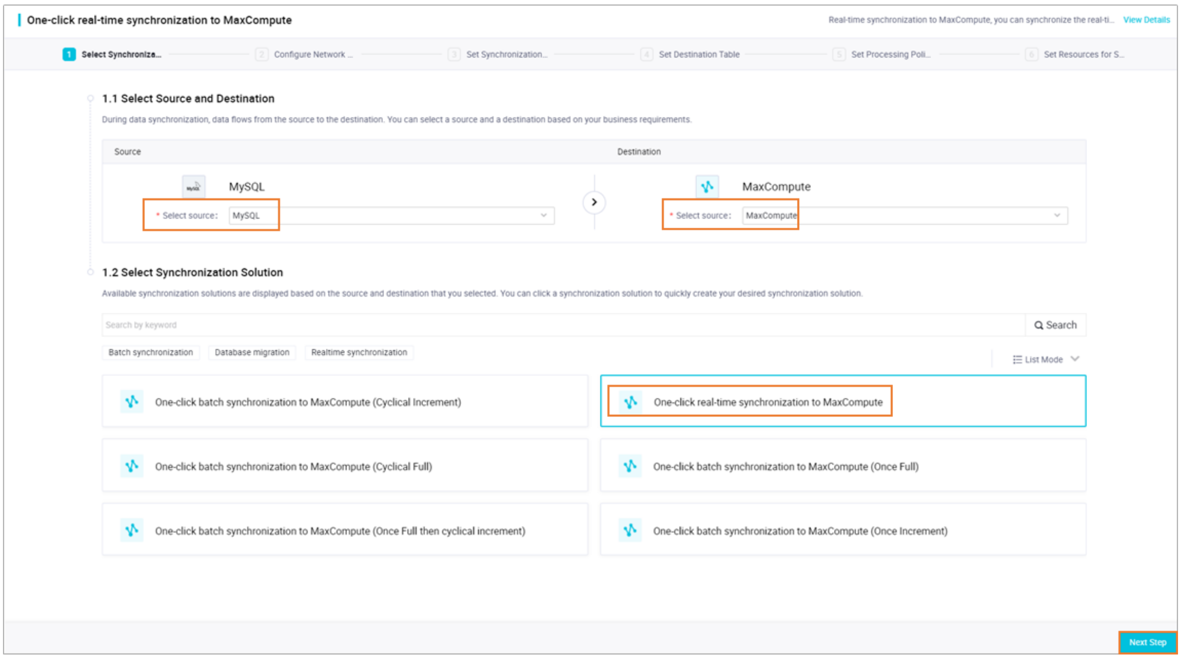
Billing

The real-time synchronization to MaxCompute synchronization solution requires periodic merging of full and incremental data. Therefore, MaxCompute computing resources are consumed. The fees for the MaxCompute computing resources are charged by MaxCompute and are positively correlated to the size of the full data and the merging cycle. For more information, see Billing method.

Create a synchronization solution

1. Log on to the DataWorks console and go to Data Integration. On the Data Integration page, click Data Synchronization Node in the left-side navigation pane. On the Tasks page, click New task to go to the Create Data Synchronization Solution page. In the Select Synchronization Solution step of this page, select a source and a destination for data synchronization from the drop-down lists. In this example, select MaxCompute as the destination type. Then, select **One-click real-time synchronization to MaxCompute** from the available synchronization solutions and click Next.

   For more information, see Select a synchronization solution.

   One-click real-time synchronization to MaxCompute

   Real-time synchronization to MaxCompute, you can synchronize the real-ti... View Details

   1 Select Synchroniza... 2 Configure Network ... 3 Set Synchronization... 4 Set Destination Table 5 Set Processing Poli... 6 Set Resources for S...

   1.1 Select Source and Destination

   During data synchronization, data flows from the source to the destination. You can select a source and a destination based on your business requirements.

Source	Destination
MySQL	MaxCompute
* Select source: MySQL	* Select source: MaxCompute

   1.2 Select Synchronization Solution

   Available synchronization solutions are displayed based on the source and destination that you selected. You can click a synchronization solution to quickly create your desired synchronization solution.

   One-click batch synchronization to MaxCompute (Cyclical Increment)

   One-click real-time synchronization to MaxCompute

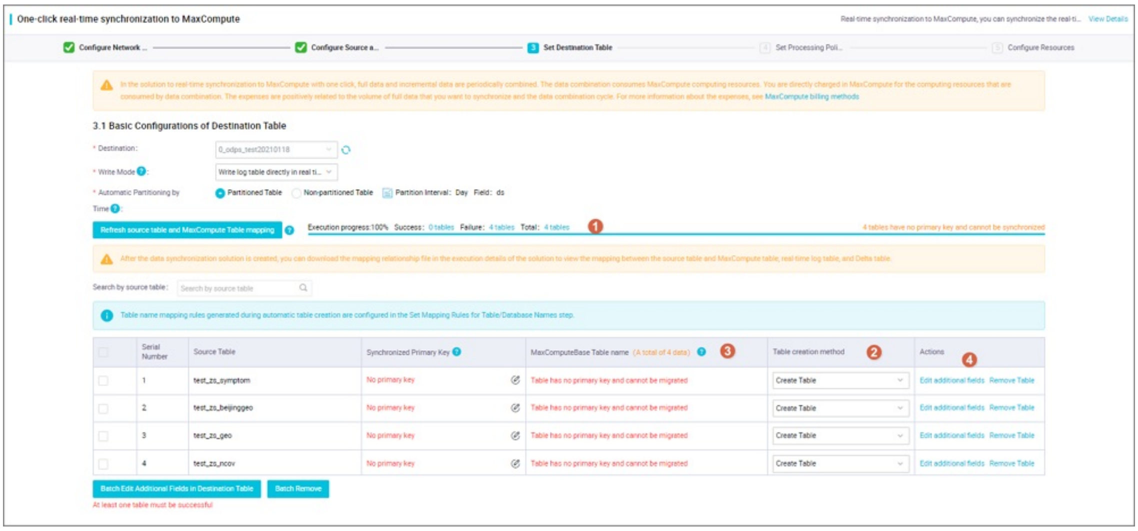
   One-click batch synchronization to MaxCompute (Cyclical Full)

   One-click batch synchronization to MaxCompute (Once Full)

   One-click batch synchronization to MaxCompute (Once Full then cyclical increment)

   One-click batch synchronization to MaxCompute (Once Increment)

2.
3.
4. Select a data source as the destination and configure formats for the destination tables.
   i. In the **Set Destination Table** step, configure the **Destination** and **Write Mode** parameters.
   ii. Click the 📝 icon next to **Automatic Partitioning by Time**. In the **Edit** dialog box, modify the partition settings for the destination tables. You can configure daily partitions. You can write data to a partitioned table or a non-partitioned table in MaxCompute.
   iii. Optional. Configure the **Batch Sync for Special Tables** parameter to specify whether to create a full batch synchronization node for tables without primary keys.
   iv. Click **Refresh source table and MaxCompute Table mapping** to create mappings between the source tables and destination MaxCompute tables.
   v. View the mapping progress, source tables, and mapped destination tables.



One-click real-time synchronization to MaxCompute

Real-time synchronization to MaxCompute, you can synchronize the real-ti... View Details

☑ Configure Network ... ☑ Configure Source a... 3 Set Destination Table 4 Set Processing Poli... 5 Configure Resources

In the solution to real-time synchronization to MaxCompute with one click, full data and incremental data are periodically combined. The data combination consumes MaxCompute computing resources. You are directly charged in MaxCompute for the computing resources that are consumed by data combination. The expenses are positively related to the volume of full data that you want to synchronize and the data combination cycle. For more information about the expenses, see MaxCompute billing methods

3.1 Basic Configurations of Destination Table

* Destination: 0_odps_test20210118

* Write Mode: Write log table directly in real ti...

* Automatic Partitioning by Time: ● Partitioned Table ○ Non-partitioned Table Partition Interval: Day Field: ds

Refresh source table and MaxCompute Table mapping

Execution progress:100% Success: 0 tables Failure: 4 tables Total: 4 tables

4 tables have no primary key and cannot be synchronized

After the data synchronization solution is created, you can download the mapping relationship file in the execution details of the solution to view the mapping between the source table and MaxCompute table, real-time log table, and Delta table.

Search by source table: Search by source table

Table name mapping rules generated during automatic table creation are configured in the Set Mapping Rules for Table/Database Names step.

□	Serial Number	Source Table	Synchronized Primary Key	MaxComputeBase Table name (A total of 4 data)	Table creation method	Actions
□	1	test_zs_symptom	No primary key	Table has no primary key and cannot be migrated	Create Table	Edit additional fields Remove Table
□	2	test_zs_beijinggeo	No primary key	Table has no primary key and cannot be migrated	Create Table	Edit additional fields Remove Table
□	3	test_zs_geo	No primary key	Table has no primary key and cannot be migrated	Create Table	Edit additional fields Remove Table
□	4	test_zs_ncov	No primary key	Table has no primary key and cannot be migrated	Create Table	Edit additional fields Remove Table

Batch Edit Additional Fields in Destination Table Batch Remove

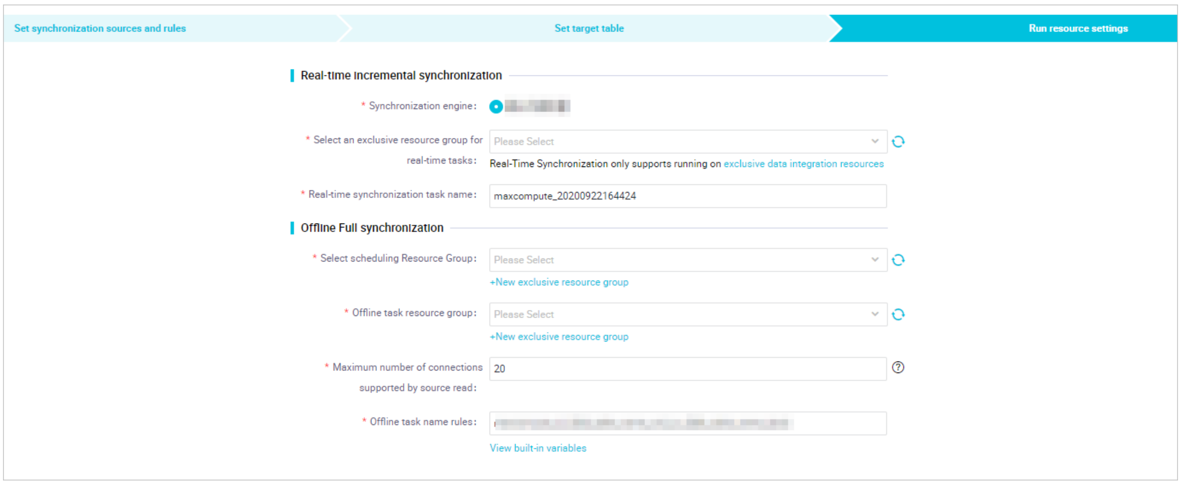
At least one table must be successful

No.	Description
1	The progress of mapping the source tables to the destination tables. Note The mapping may require a long period of time if data is synchronized from a large number of tables.
2	The method used to create a destination table. Valid values: **Create Table** and **Use Existing Table**.
3	The name of the destination table. The table name that appears in the MaxComputeBase Table name column varies based on the method that you select from the drop-down list in the **Table creation method** column. ▪ If you select **Create Table** from the drop-down list in the Table creation method column, the name of the destination table that is automatically created appears. You can click the table name to view and modify the table creation statement. ▪ If you select **Use Existing Table** from the drop-down list in the Table creation method column, you must select a table name from the drop-down list in the MaxComputeBase Table name column. Note If a source table does not contain the primary key, you can click the edit icon next to No primary key in the Synchronized Primary Key column and specify the primary key for the source table so that full and incremental data can be synchronized from the source table.
4	You can click **Edit additional fields** in the Actions column to add additional fields to the destination table in addition to the fields in the source table. Note If you select **Create Table** from the drop-down list in the Table creation method column and specify additional fields, the related columns are automatically added to the destination table. If you select Use Existing Table from the drop-down list in the Table creation method column and you want to add additional fields to the existing destination table, you must make sure that the related columns already exist in the destination table. This way, data can be written to the columns. DataWorks does not modify the schema of the existing table to add new columns to the existing table.

vi. Click **Next**.

5. Configure the resources required by the synchronization solution.

   In the **Configure Resources** step, configure the parameters.



Set synchronization sources and rules | Set target table | Run resource settings

Real-time incremental synchronization

* Synchronization engine: ● [redacted]

* Select an exclusive resource group for real-time tasks: Please Select

Real-Time Synchronization only supports running on exclusive data integration resources

* Real-time synchronization task name: maxcompute_20200922164424

Offline Full synchronization

* Select scheduling Resource Group: Please Select

+New exclusive resource group

* Offline task resource group: Please Select

+New exclusive resource group

* Maximum number of connections supported by source read: 20

* Offline task name rules: [redacted]

View built-in variables

Parameter	Description
Synchronization engine	The compute engine used for data synchronization. Default value: **Default Embedded Engine**.
Select an exclusive resource group for real-time tasks	The exclusive resource group used to run the real-time synchronization node generated by the synchronization solution. Select an exclusive group from the drop-down list. Note You can use only exclusive resource groups for Data Integration to run real-time synchronization nodes. For more information, see Create and use an exclusive resource group for Data Integration.
Real-time synchronization task name	The name of the real-time synchronization node.
Resource Group for Scheduling	The exclusive resource group used to schedule the batch synchronization node generated by the synchronization solution. You can use only exclusive resource groups for Data Integration to run data synchronization solutions. For more information about exclusive resource groups for Data Integration, see Create and use an exclusive resource group for Data Integration.
Exclusive Resource Groups for Full Batch Sync Nodes	
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source database.
Offline task name rules	The name of the batch synchronization node that is used to synchronize the full data of the source. After a synchronization solution is created, DataWorks first generates a batch synchronization node to synchronize full data, and then generates a real-time synchronization node to synchronize incremental data.

6. Click **Complete Configuration**. The synchronization solution is configured.

Run the synchronization solution

On the **Solution task list** tab of the Tasks page, find the created synchronization solution and click **Submit and Run** in the Operation column to run the synchronization solution.

4.7.3. Synchronize full data to MaxCompute on a regular basis

After you configure data sources, network environments, and resource groups, you can create and run sync nodes. This topic describes how to configure a sync solution to synchronize full data to MaxCompute on a regular basis and view the status of the nodes generated by the sync solution.

Prerequisites

Before you create a synchronization solution, make sure that the following operations are performed:

- Plan and configure resources
- Configure a source PolarDB data source
- Configure data sources for data synchronization from Oracle
- Configure data sources for data synchronization from MySQL
- Add a data source

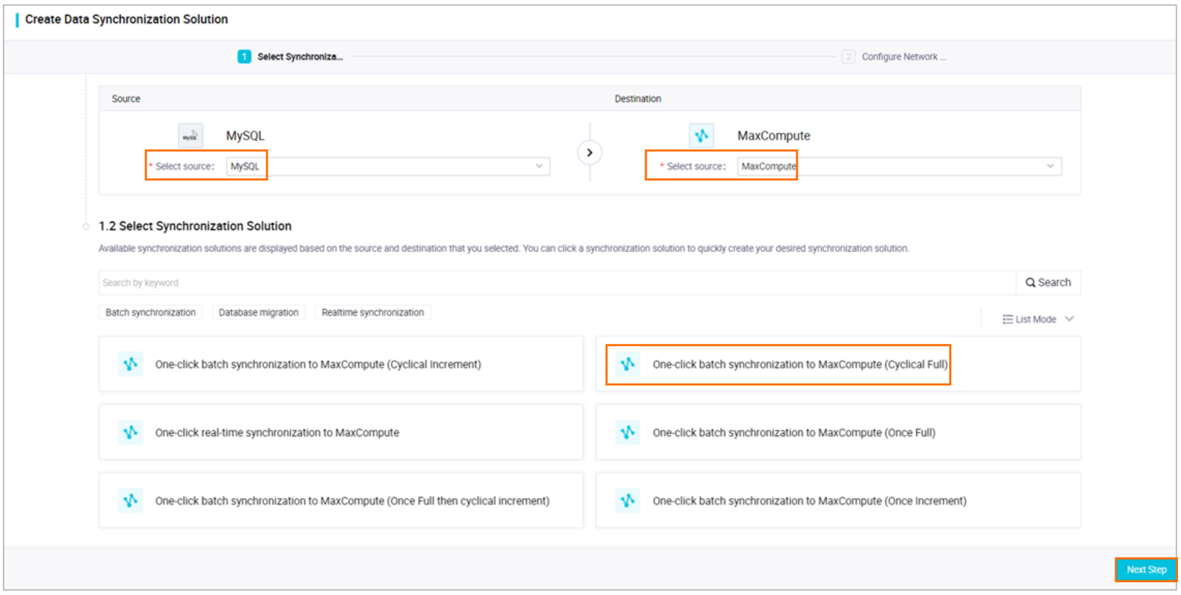
Scenarios

The One-click batch synchronization to MaxCompute (Cyclical Full) solution is applicable to the scenarios in which you need to synchronize full data from specific tables to MaxCompute on a regular basis. This solution is suitable for periodic data synchronization from a large number of tables. You can synchronize source tables in batches to reduce the load. The scheduling time is flexible with many options to facilitate periodic data synchronization.

Configure a sync solution

1. Go to the Create Data Synchronization Solution wizard. Select the source and the destination for data synchronization from the drop-down lists. In this scenario, select MaxCompute as the destination. After that, select **One-click batch synchronization to MaxCompute (Cyclical Full)** from the available sync solutions.

   For more information, see Select a synchronization solution.



Create Data Synchronization Solution

Source: MySQL
* Select source: MySQL

Destination: MaxCompute
* Select source: MaxCompute

1.2 Select Synchronization Solution

Available synchronization solutions are displayed based on the source and destination that you selected. You can click a synchronization solution to quickly create your desired synchronization solution.

- One-click batch synchronization to MaxCompute (Cyclical Increment)
- One-click batch synchronization to MaxCompute (Cyclical Full)
- One-click real-time synchronization to MaxCompute
- One-click batch synchronization to MaxCompute (Once Full)
- One-click batch synchronization to MaxCompute (Once Full then cyclical increment)
- One-click batch synchronization to MaxCompute (Once Increment)

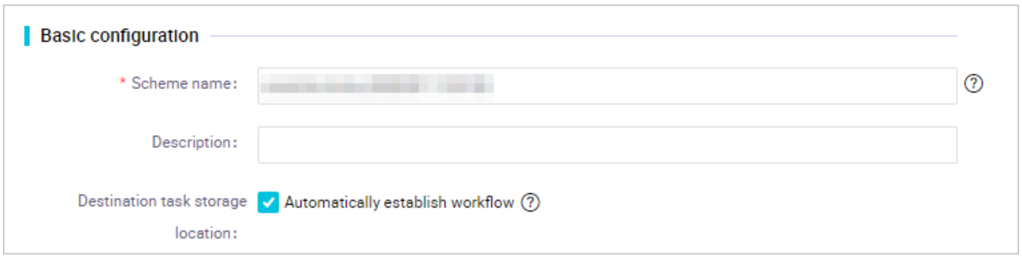
2. Configure network connection for data synchronization.

   Select the data source, exclusive resource group for Data Integration, and destination data source as prompted, and then test the network connectivity. After that, click **Next Step**. You must prepare the exclusive resource groups and the network connection solution that you want to use. In addition, you must create connections to data sources in DataWorks and configure network connectivity as required, such as a whitelist. This avoids failures in connectivity tests. For more information, see Plan and configure resources.

3. Configure the source and rules for data synchronization.

i. Configure the basic information.

In the **Basic Configuration** section, set the parameters that are described in the following table.



Basic configuration

* Scheme name: [redacted]

Description:

Destination task storage location: ☑ Automatically establish workflow

Parameter	Description
Solution Name	The name of the sync solution. The name can be a maximum of 50 characters in length.
Description	The description of the sync solution. The description can be a maximum of 50 characters in length.
Location	If you select Automatic Workflow Creation, DataWorks automatically creates a workflow named in the format of clone_database_Source name+to+Destination name. All sync nodes generated by the sync solution are placed in the **Data Integration** folder of this workflow. If you clear **Automatic Workflow Creation**, you must select a directory from the **Select Location** drop-down list. All sync nodes generated by the sync solution are placed in the specified directory.

ii. Check the data source information.

The information about the data source selected in the preceding step is displayed in the Data Source section, and the encoding type is specified by default. You must check the information and determine whether to change the encoding type.

iii. Select the source tables for synchronization.

Select the tables whose data you want to synchronize to MaxCompute as prompted. After you select the source tables, data is synchronized from the selected tables to MaxCompute based on the configuration of this sync solution.

Notice If a selected table does not have a primary key, the table data cannot be synchronized in real time.

iv. Configure mapping rules for the names of the source and destination tables.

Click **Add rule**, select a rule type, and then configure the mapping rules. Supported rule types are **Conversion Rule for Table Name** and **Rule for Destination Table name**.

- **Conversion Rule for Table Name**: the rule used to convert the names of source tables to those of destination tables.
- **Rule for Destination Table name**: the rule used to add a prefix or a suffix to the converted names of destination tables.

v. Click **Next Step**.

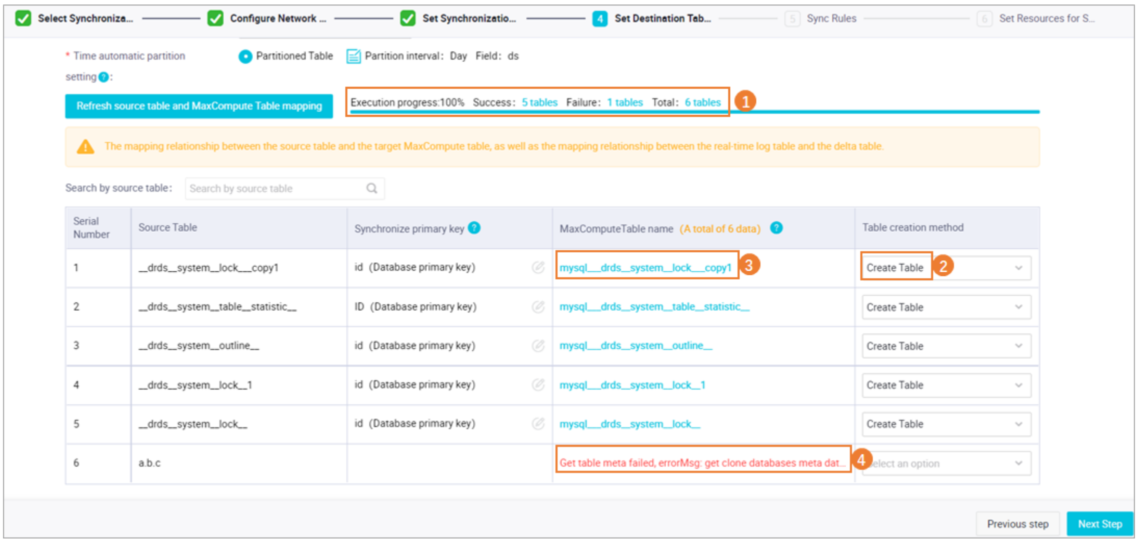
4. Configure the destination.

i. The destination data source selected in the preceding step is displayed on the page. Check whether the displayed information is valid.

ii. Click the icon next to **Time automatic partition setting**. In the **Edit** dialog box, modify the partition settings for the destination tables. You can configure daily partitions.

iii. Click **Refresh source table and MaxCompute Table mapping** to create the mappings between the source tables and destination MaxCompute tables.

iv. View the mapping progress, source tables, and mapped destination tables.



* Time automatic partition setting: ● Partitioned Table Partition interval: Day Field: ds

Execution progress:100% Success: 5 tables Failure: 1 tables Total: 6 tables

The mapping relationship between the source table and the target MaxCompute table, as well as the mapping relationship between the real-time log table and the delta table.

Search by source table: Search by source table

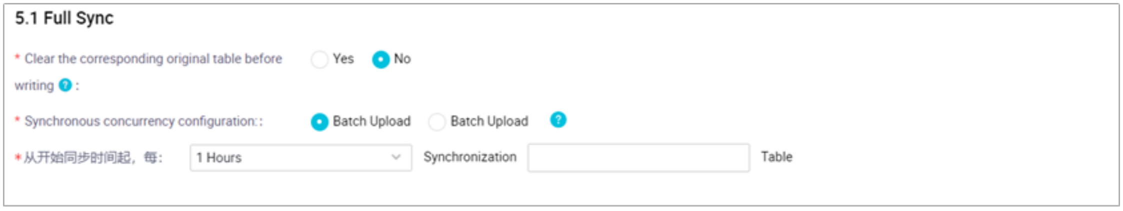
Serial Number	Source Table	Synchronize primary key	MaxComputeTable name (A total of 6 data)	Table creation method
1	__drds__system__lock__copy1	id (Database primary key)	mysql__drds__system__lock__copy1	Create Table
2	__drds__system__table__statistic__	ID (Database primary key)	mysql__drds__system__table__statistic__	Create Table
3	__drds__system__outline__	id (Database primary key)	mysql__drds__system__outline__	Create Table
4	__drds__system__lock__1	id (Database primary key)	mysql__drds__system__lock__1	Create Table
5	__drds__system__lock__	id (Database primary key)	mysql__drds__system__lock__	Create Table
6	a.b.c		Get table meta failed, errorMsg: get clone databases meta dat...	Select an option

No.	Description
1	The progress of mapping the source tables to the destination tables. Note The mapping may require a long period of time if you want to synchronize data from a large number of tables.
2	The source of the destination table. Valid values: **Create Table** and **Use Existing Table**.
3	The name of the destination table. The table name that appears varies based on the value that you selected from the drop-down list in the **Table creation method** column. - If you set the **Table creation method** parameter to **Create Table**, the name of the destination table that is automatically created appears. You can click the table name to view and modify the table creation statements. - If you set the **Table creation method** parameter to **Use Existing Table**, you must select a table name from the drop-down list in the MaxComputeTable name column.
4	If a source table does not have a primary key, an error message appears to remind you that the current source table does not have a primary key and cannot be synchronized. The synchronization can be performed if one of the selected source tables has a primary key. Source tables without primary keys are ignored during the synchronization.

v. Click **Next Step**.

5. Configure synchronization rules.

i. Configure rules for full data synchronization.



5.1 Full Sync

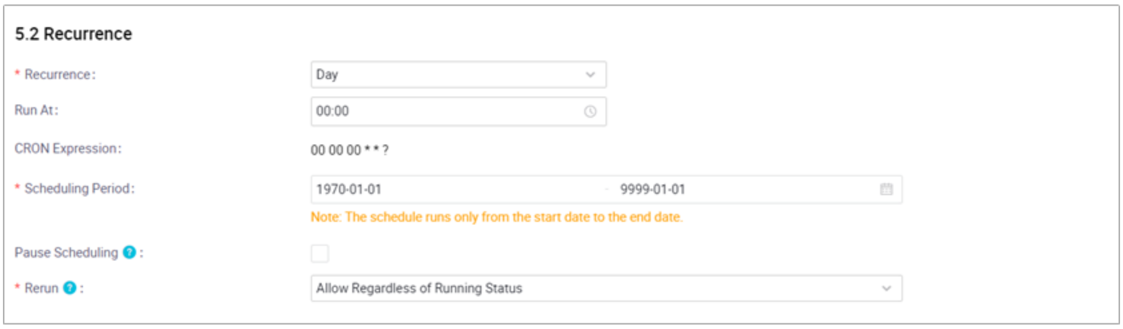
* Clear the corresponding original table before writing ? : ○ Yes ● No

* Synchronous concurrency configuration:: ● Batch Upload ○ Batch Upload ?

*从开始同步时间起，每: 1 Hours Synchronization Table

Parameter	Description
Clear the corresponding original table before writing	Enable this feature as needed. If you enable this feature, the previously synchronized tables are deleted from MaxCompute each time before data is synchronized. We recommend that you enable this feature with caution.
Synchronous concurrency configuration	You can specify whether to synchronize source tables in batches or at a time. We recommend that you select **Batch Upload** to synchronize a large number of source tables in batches. This prevents a heavy load from affecting data synchronization.
Interval for batch upload Specify the number of tables to be synchronized after each interval of time.	If you set the **Synchronous concurrency configuration** parameter to **Batch Upload**, you must specify the number of tables to be synchronized after each interval of time. The interval can be at least 15 minutes or several hours. For example, you can set the scheduling time to 05:00 every day, and 300 source tables are to be synchronized. Data synchronization can last for a maximum of 19 hours from 05:00 to the end of the same day based on the recurrence. To prevent a heavy load, you can divide 300 source tables into six batches and set the interval to three hours. This way, data synchronization starts at 05:00, and data is synchronized from 50 tables for each batch every three hours. **Note** You need to set the interval for batch upload based on the recurrence of data synchronization. The sum of the intervals for batch upload must be less than the available duration for data synchronization. In the preceding example, six batches of source tables are synchronized to MaxCompute every three hours. Therefore, the sum of the intervals for batch upload is 18 hours, which is less than 19 hours, the available duration for data synchronization.

ii. Configure the recurrence for data synchronization.



5.2 Recurrence

* Recurrence: Day

Run At: 00:00

CRON Expression: 00 00 00 * * ?

* Scheduling Period: 1970-01-01 9999-01-01

Note: The schedule runs only from the start date to the end date.

Pause Scheduling ? : □

* Rerun ? : Allow Regardless of Running Status

Set the parameters to configure the recurrence of data synchronization as needed, such as the **Recurrence**, **Run At**, and **Scheduling Period** parameters. The configuration of the scheduling parameters is similar to that for a regular node. For more information, see Configure time properties.

i. Click **Next Step**.

6. Configure the resources required by the sync solution.

In the **Set Resources for Solution Running** step, check the name of the sync node to be generated by the sync solution, the resource group for Data Integration, and the resource group for scheduling. Then, set the **Maximum number of connections supported by source read** parameter.

> Note The **Maximum number of connections supported by source read** parameter specifies the maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. You must set this parameter based on the capabilities of the source. If the specified number of connections is beyond the capabilities of the source, the data may not be read from the source.

7. Click **Complete Configuration**. The sync solution is configured.

Run the synchronization solution

On the **Solution task list** tab of the Tasks page, find the created synchronization solution and click **Submit and Run** in the Operation column to run the synchronization solution.

4.7.4. Synchronize incremental data to MaxCompute on a regular basis

After you configure data sources, network environments, and resource groups, you can create and run sync nodes. This topic describes how to configure a sync solution to synchronize incremental data to MaxCompute on a regular basis and view the status of the nodes generated by the sync solution.

Prerequisites

Before you create a synchronization solution, make sure that the following operations are performed:

- Plan and configure resources
- Configure a source PolarDB data source
- Configure data sources for data synchronization from Oracle
- Configure data sources for data synchronization from MySQL
- Add a data source

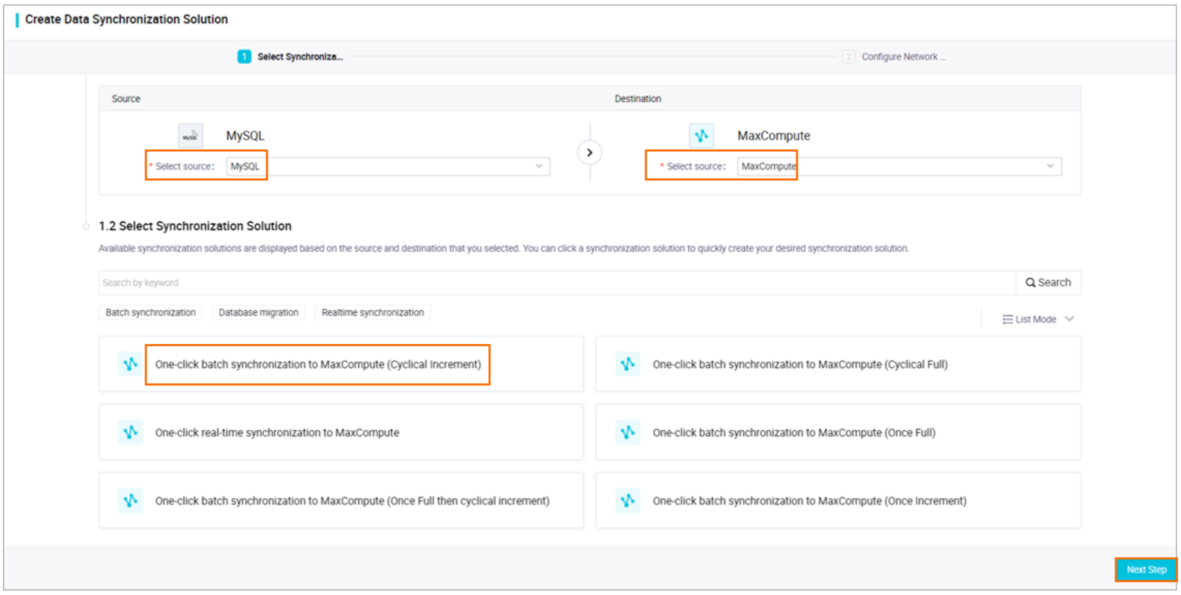
Scenarios

The One-click batch synchronization to MaxCompute (Cyclical Increment) solution is applicable to the scenarios in which you need to synchronize incremental data from specific tables to MaxCompute on a regular basis. You can use the SQL WHERE clause to extract incremental data from the source tables and synchronize the data to MaxCompute. The scheduling time is flexible with many options to facilitate periodic data synchronization.

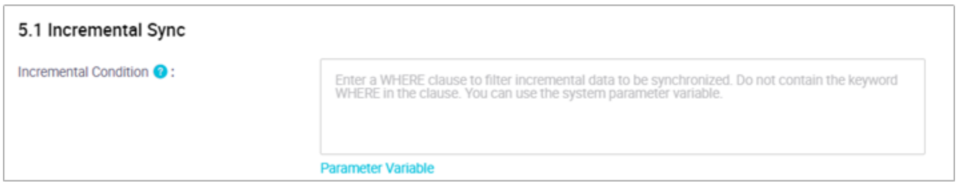
Configure a sync solution

1. Go to the Create Data Synchronization Solution wizard. Select the source and the destination for data synchronization from the drop-down lists. In this scenario, select MaxCompute as the destination. After that, select **One-click batch synchronization to MaxCompute (Cyclical Increment)** from the available sync solutions.

   For more information, see Select a synchronization solution.


Create Data Synchronization Solution
1 Select Synchroniza...
2 Configure Network ...
Source
Destination
MySQL
* Select source: MySQL
MaxCompute
* Select source: MaxCompute
1.2 Select Synchronization Solution
Available synchronization solutions are displayed based on the source and destination that you selected. You can click a synchronization solution to quickly create your desired synchronization solution.
Search by keyword
Search
Batch synchronization
Database migration
Realtime synchronization
List Mode
One-click batch synchronization to MaxCompute (Cyclical Increment)
One-click batch synchronization to MaxCompute (Cyclical Full)
One-click real-time synchronization to MaxCompute
One-click batch synchronization to MaxCompute (Once Full)
One-click batch synchronization to MaxCompute (Once Full then cyclical increment)
One-click batch synchronization to MaxCompute (Once Increment)
Next Step

2.
3.
4.
5. Configure synchronization rules.
   i. Configure rules for incremental data synchronization.


5.1 Incremental Sync
Incremental Condition ? :
Enter a WHERE clause to filter incremental data to be synchronized. Do not contain the keyword WHERE in the clause. You can use the system parameter variable.
Parameter Variable

You can use the SQL WHERE clause to extract incremental data from the source tables. You need only to enter the WHERE clause in the Incremental Condition field without the need to enter the WHERE keyword. In addition, you can use the built-in system variables to write the WHERE clause. For example, the `${bizdate}` variable specifies the data timestamp, and the `${cyctime}` variable specifies the scheduling time. For more information, see Overview of scheduling parameters.

   ii.
   iii.

6.
7.

Run the synchronization solution

On the **Solution task list** tab of the Tasks page, find the created synchronization solution and click **Submit and Run** in the Operation column to run the synchronization solution.

4.7.5. Synchronize full data to MaxCompute at a time

The One-click batch synchronization to MaxCompute (Once Full) solution is applicable to the scenarios in which you need to synchronize full data from specific tables to MaxCompute at a time. After you configure data sources, network environments, and resource groups, you can create and run sync nodes. This topic describes how to configure a sync solution to synchronize full data to MaxCompute at a time and view the status of the nodes generated by the sync solution.

Prerequisites

Before you create a synchronization solution, make sure that the following operations are performed:

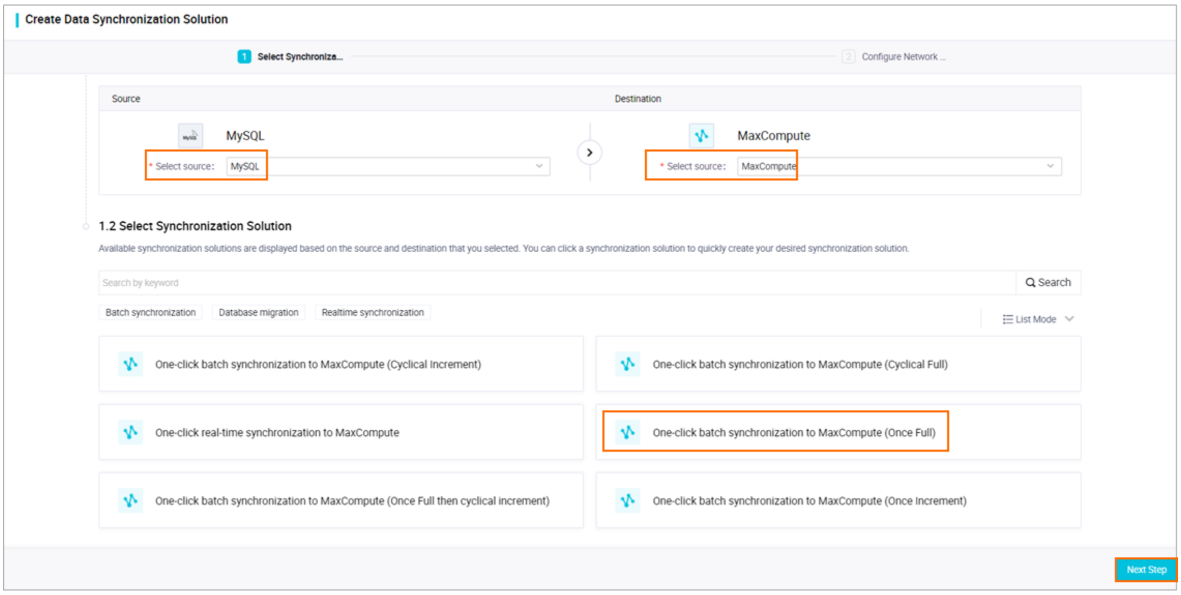
- Plan and configure resources

- Configure a source PolarDB data source
- Configure data sources for data synchronization from Oracle
- Configure data sources for data synchronization from MySQL
- Add a data source

Configure a sync solution

1. Go to the Create Data Synchronization Solution wizard. Select the source and the destination for data synchronization from the drop-down lists. In this scenario, select MaxCompute as the destination. After that, select **One-click batch synchronization to MaxCompute (Once Full)** from the available sync solutions.

   For more information, see Select a synchronization solution.



Create Data Synchronization Solution

1 Select Synchroniza... 2 Configure Network ...

Source: MySQL — * Select source: MySQL

Destination: MaxCompute — * Select source: MaxCompute

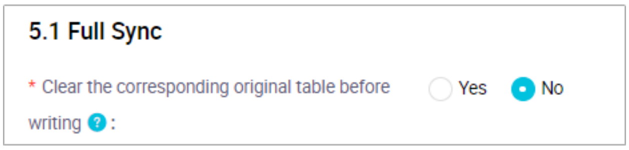
1.2 Select Synchronization Solution

Available synchronization solutions are displayed based on the source and destination that you selected. You can click a synchronization solution to quickly create your desired synchronization solution.

Search by keyword
Search
Batch synchronization
Database migration
Realtime synchronization
List Mode
One-click batch synchronization to MaxCompute (Cyclical Increment)
One-click batch synchronization to MaxCompute (Cyclical Full)
One-click real-time synchronization to MaxCompute
One-click batch synchronization to MaxCompute (Once Full)
One-click batch synchronization to MaxCompute (Once Full then cyclical increment)
One-click batch synchronization to MaxCompute (Once Increment)
Next Step

2.
3.
4.
5. Configure synchronization rules.

   i. Configure rules for full data synchronization.



5.1 Full Sync

* Clear the corresponding original table before writing ? : ○ Yes ● No

Parameter	Description
Clear the corresponding original table before writing	Enable this feature as needed. If you enable this feature, the previously synchronized tables are deleted from MaxCompute each time before data is synchronized. We recommend that you enable this feature with caution.

   ii.

6.
7.

Run the synchronization solution

On the **Solution task list** tab of the Tasks page, find the created synchronization solution and click **Submit and Run** in the Operation column to run the synchronization solution.

4.7.6. Synchronize incremental data to MaxCompute at a time

The One-click batch synchronization to MaxCompute (Once Increment) solution is applicable to the scenarios in which you need to synchronize incremental data from specific tables to MaxCompute at a time. You can use the SQL WHERE clause to extract incremental data from the source tables. After you configure data sources, network environments, and resource groups, you can create and run sync nodes. This topic describes how to configure a sync solution to synchronize incremental data to MaxCompute at a time and view the status of the nodes generated by the sync solution.

Prerequisites

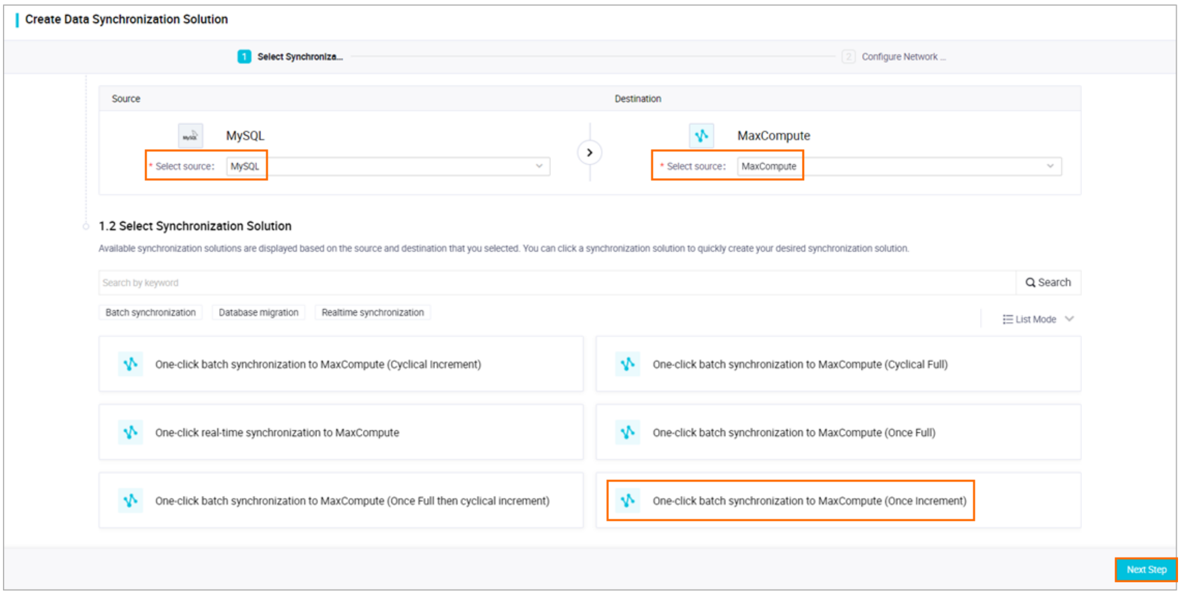
Before you create a synchronization solution, make sure that the following operations are performed:

- Plan and configure resources
- Configure a source PolarDB data source
- Configure data sources for data synchronization from Oracle
- Configure data sources for data synchronization from MySQL
- Add a data source

Configure a sync solution

1. Go to the Create Data Synchronization Solution wizard. Select the source and the destination for data synchronization from the drop-down lists. In this scenario, select MaxCompute as the destination. After that, select **One-click batch synchronization to MaxCompute (Once Increment)** from the available sync solutions.

   For more information, see Select a synchronization solution.

   Create Data Synchronization Solution

   Source: MySQL, * Select source: MySQL

   Destination: MaxCompute, * Select source: MaxCompute

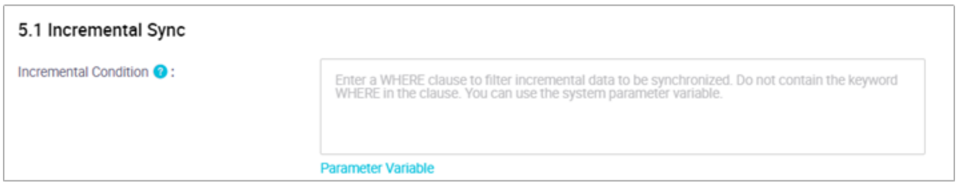
   1.2 Select Synchronization Solution

   Available synchronization solutions are displayed based on the source and destination that you selected. You can click a synchronization solution to quickly create your desired synchronization solution.

   1 Select Synchroniza...
2 Configure Network ...
Search by keyword
Search
Batch synchronization
Database migration
Realtime synchronization
List Mode
One-click batch synchronization to MaxCompute (Cyclical Increment)
One-click batch synchronization to MaxCompute (Cyclical Full)
One-click real-time synchronization to MaxCompute
One-click batch synchronization to MaxCompute (Once Full)
One-click batch synchronization to MaxCompute (Once Full then cyclical increment)
One-click batch synchronization to MaxCompute (Once Increment)
Next Step

2.
3.
4.
5. Configure synchronization rules.

i. Configure rules for incremental data synchronization.



5.1 Incremental Sync

Incremental Condition :

Enter a WHERE clause to filter incremental data to be synchronized. Do not contain the keyword WHERE in the clause. You can use the system parameter variable.

You can use the SQL WHERE clause to extract incremental data from the source tables. You need only to enter the WHERE clause in the Incremental Condition field without the need to enter the WHERE keyword. In addition, you can use the built-in system variables to write the WHERE clause. For example, the `${bizdate}` variable specifies the data timestamp, and the `${cyctime}` variable specifies the scheduling time. For more information, see Overview of scheduling parameters.

ii.

6.

7.

Run the synchronization solution

On the **Solution task list** tab of the Tasks page, find the created synchronization solution and click **Submit and Run** in the Operation column to run the synchronization solution.

4.7.7. Synchronize full data to MaxCompute at a time and then synchronize incremental data on a regular basis

The One-click batch synchronization to MaxCompute (Once Full then cyclical increment) solution is applicable to the scenarios in which you need to synchronize full data from specific tables to MaxCompute at a time and then synchronize incremental data from these tables on a regular basis. After you configure data sources, network environments, and resource groups, you can create and run synchronization nodes. This topic describes how to configure a synchronization solution to synchronize full data to MaxCompute at a time and then synchronize incremental data on a regular basis, and view the status of the nodes generated by the synchronization solution.

Prerequisites

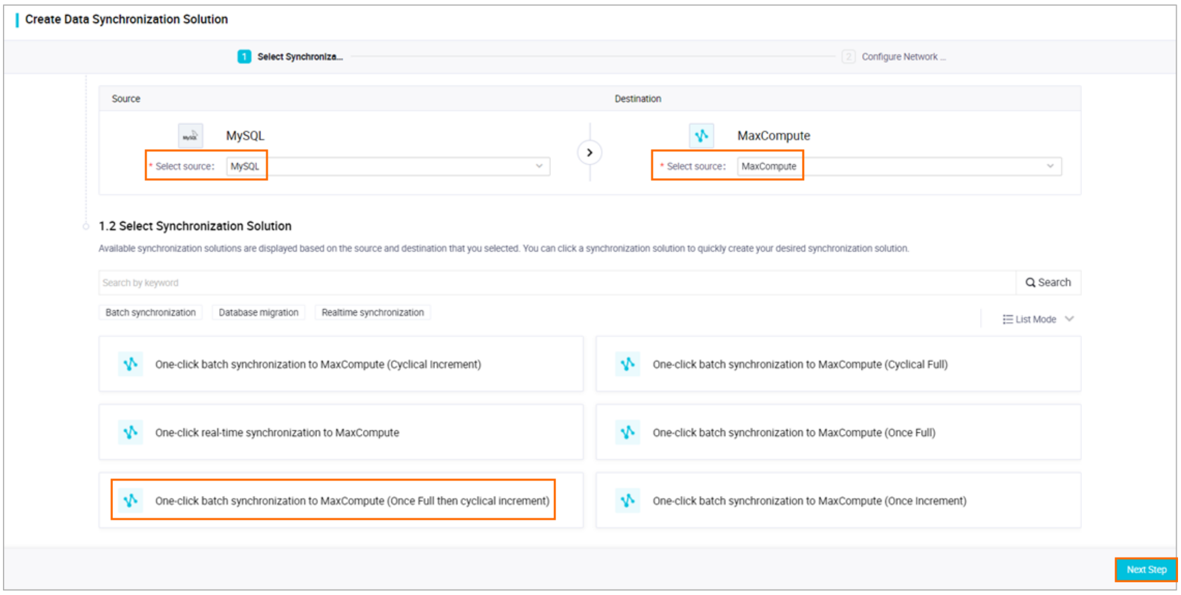
Before you create a synchronization solution, make sure that the following operations are performed:

- Plan and configure resources
- Configure a source PolarDB data source
- Configure data sources for data synchronization from Oracle
- Configure data sources for data synchronization from MySQL
- Add a data source

Create a synchronization solution

1. Go to the Create Data Synchronization Solution wizard. Select the source and the destination for data synchronization from the drop-down lists. In this scenario, select MaxCompute as the destination. After that, select **One-click batch synchronization to MaxCompute (Once Full then cyclical increment)** from the available synchronization solutions.

   For more information, see Select a synchronization solution.



Create Data Synchronization Solution

1 Select Synchroniza... — 2 Configure Network ...

Source: MySQL — * Select source: MySQL

Destination: MaxCompute — * Select source: MaxCompute

1.2 Select Synchronization Solution

Available synchronization solutions are displayed based on the source and destination that you selected. You can click a synchronization solution to quickly create your desired synchronization solution.

- One-click batch synchronization to MaxCompute (Cyclical Increment)
- One-click batch synchronization to MaxCompute (Cyclical Full)
- One-click real-time synchronization to MaxCompute
- One-click batch synchronization to MaxCompute (Once Full)
- One-click batch synchronization to MaxCompute (Once Full then cyclical increment)
- One-click batch synchronization to MaxCompute (Once Increment)

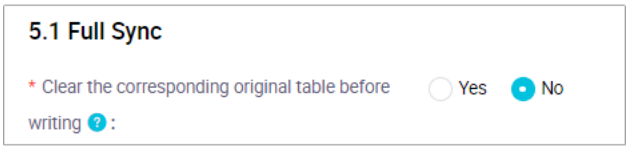
2.

3.

4.

5. Configure synchronization rules.

   i. Configure rules for full data synchronization.

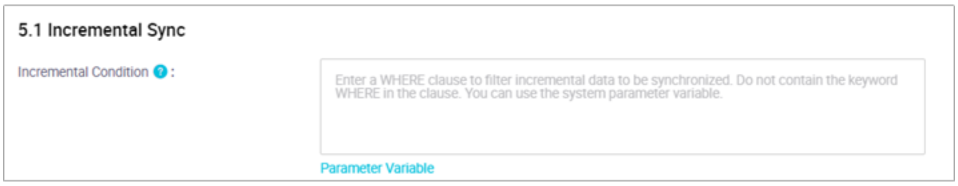


5.1 Full Sync

* Clear the corresponding original table before writing : ○ Yes ● No

Parameter	Description
Clear the corresponding original table before writing	Enable this feature as needed. If you enable this feature, the previously synchronized tables are deleted from MaxCompute each time before data is synchronized. We recommend that you enable this feature with caution.

   ii. Configure rules for incremental data synchronization.



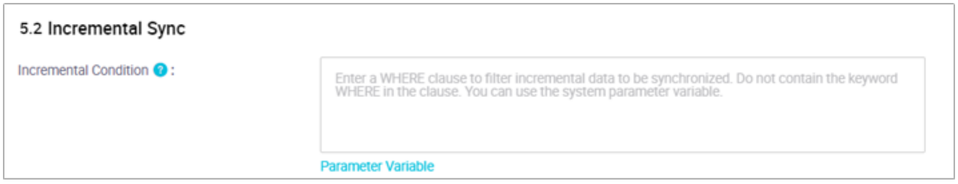
5.1 Incremental Sync

Incremental Condition :

Enter a WHERE clause to filter incremental data to be synchronized. Do not contain the keyword WHERE in the clause. You can use the system parameter variable.
Parameter Variable

You can use the SQL WHERE clause to extract incremental data from the source tables. You need only to enter the WHERE clause in the Incremental Condition field without the need to enter the WHERE keyword. In addition, you can use the built-in system variables to write the WHERE clause. For example, the `${bizdate}` variable specifies the data timestamp, and the `${cyctime}` variable specifies the scheduling time. For more information, see Overview of scheduling parameters.

iii. Configure rules for incremental data synchronization.



5.2 Incremental Sync

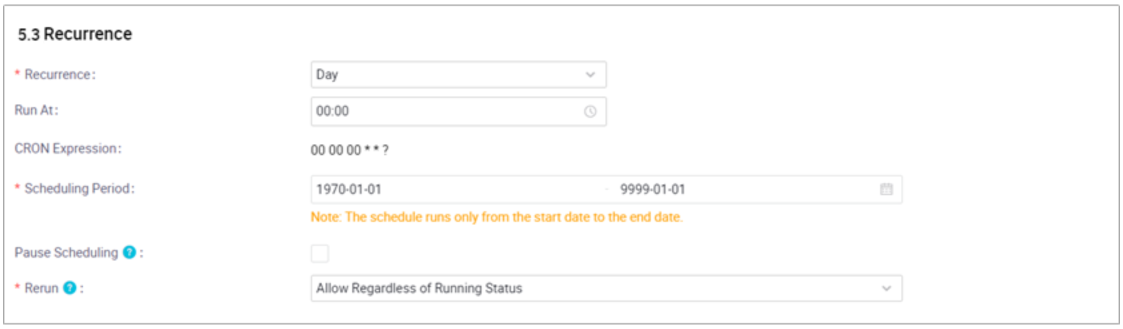
Incremental Condition ? :

Enter a WHERE clause to filter incremental data to be synchronized. Do not contain the keyword WHERE in the clause. You can use the system parameter variable.

Parameter Variable

You can use the SQL WHERE clause to extract incremental data from the source tables. You need only to enter the WHERE clause in the Condition for Incremental Synchronization field without the need to enter the WHERE keyword. In addition, you can use the built-in system variables to write the WHERE clause. For example, the `${bizdate}` variable specifies the data timestamp, and the `${cyctime}` variable specifies the scheduling time. For more information, see Overview of scheduling parameters.

iv. Configure the recurrence for data synchronization.



5.3 Recurrence

* Recurrence: Day

Run At: 00:00

CRON Expression: 00 00 00 * * ?

* Scheduling Period: 1970-01-01 9999-01-01

Note: The schedule runs only from the start date to the end date.

Pause Scheduling ? : □

* Rerun ? : Allow Regardless of Running Status

Set the parameters to configure the recurrence of data synchronization as needed, such as the **Recurrence**, **Run At**, and **Scheduling Period** parameters. The configuration of the scheduling parameters is similar to that for a regular node. For more information, see Configure time properties.

v.

6.

7.

Run the synchronization solution

On the **Solution task list** tab of the Tasks page, find the created synchronization solution and click **Submit and Run** in the Operation column to run the synchronization solution.

4.8. Synchronize data to Kafka

4.8.1. Plan and configure resources

When you use DataWorks to synchronize data, you can use only exclusive resource groups for Data Integration to run Data Integration nodes. In addition, you can select a shared or exclusive resource group for scheduling based on your business requirements. This topic describes the resources that are used for data synchronization and how to configure the resources.

Context

- Resource planning and preparation

  When you synchronize data, Data Integration nodes are run based on resource groups for Data Integration and resource groups for scheduling. You can use only exclusive resource groups for Data Integration. Before you synchronize data, you must purchase an exclusive resource group for Data Integration and add this exclusive resource group to your DataWorks workspace.

  For more information about exclusive resource groups for Data Integration, see Exclusive resources for Data Integration.

- Network connections

  An exclusive resource group for Data Integration is essentially a group of ECS instances. After you purchase such an exclusive resource group, it is isolated from other services. You must associate the resource group with a virtual private cloud (VPC) to ensure network connectivity between the resource group and data sources during subsequent data synchronization.

What's next

After you plan and configure resources, you can configure data sources. You must connect the exclusive resource group for Data Integration to the source and destination data sources. You must also create an account and grant the required permissions to the account. This account is used to access the source and destination data sources. The preceding operations help you create a sync node. For more information about how to configure source data sources, see Configure a source MySQL data source, Configure a source Oracle data source, and Configure a source PolarDB data source.

4.8.2. Configure a source MySQL data source

Before you use DataWorks to synchronize data from a MySQL data source to Kafka in real time, you can refer to the operations in this topic to configure the network, whitelists, and permissions for the data sources to implement data synchronization.

Prerequisites

Before you configure the source data source, make sure that the following operations are performed:

- Prepare data sources: A source MySQL data source and a destination Kafka data source are prepared.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
    - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
    - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.
- Execute the following statement to check the version of your MySQL database. You can use a `MySQL 5.X` or `MySQL 8.X` database.

```
select version();
```

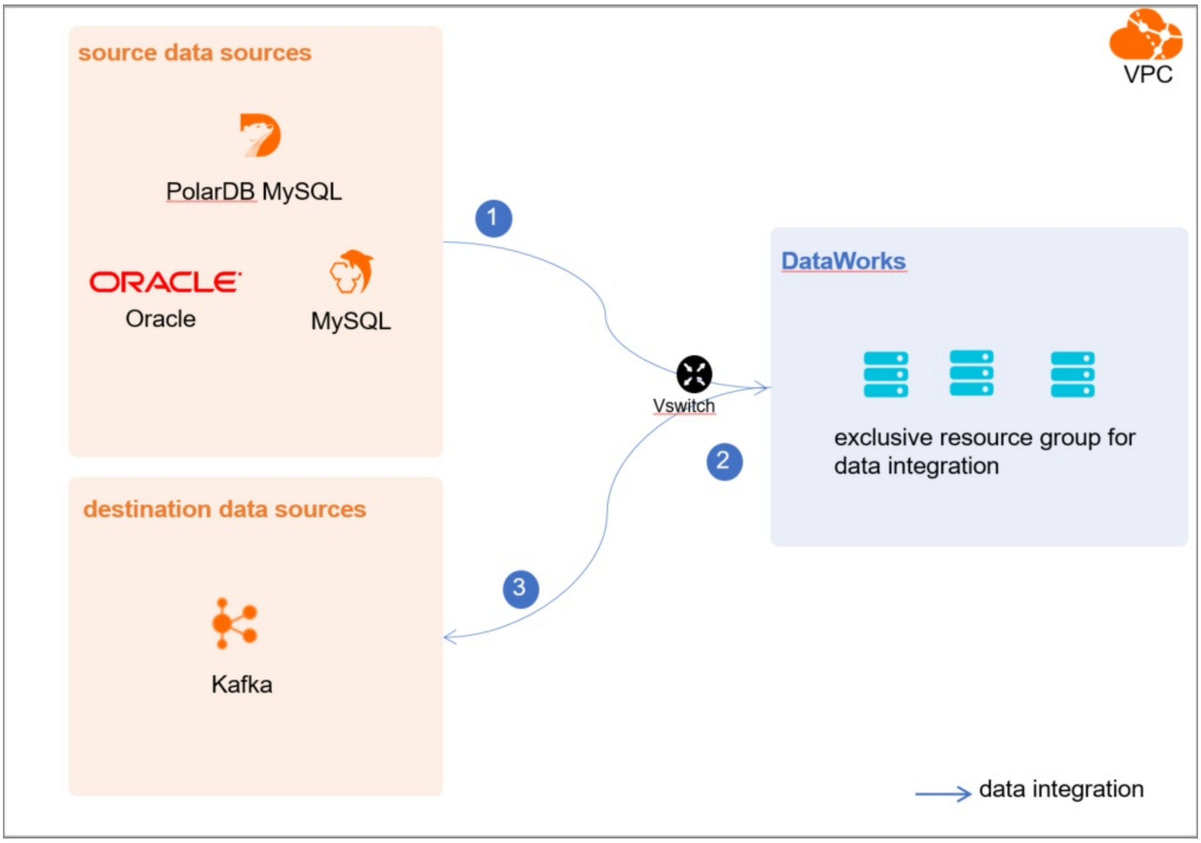
> **Note** Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.X` or `V8.X`. PolarDB for MySQL is not supported. If the MySQL version of your ApsaraDB RDS for MySQL database is not `V5.X` or `V8.X`, use an ApsaraDB RDS for MySQL database whose MySQL version is `V5.X` or `V8.X`. Otherwise, the data synchronization node fails to run.

Context

Before you synchronize data from the source data source to the destination data source, make sure that the data sources are connected to an exclusive resource group for Data Integration. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for data sources

If the source and destination data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source data source is a MySQL database, you must enable the binary logging feature. Binary logs record changes to all table schemas and modifications on table data. You can execute statements such as CREATE and ALTER to perform operations on table schemas. You can execute statements such as INSERT, UPDATE, and DELETE to perform operations on table data. You can use binary logs to view the change history of the database, back up incremental data and restore data in the database, and replicate data from the primary database to secondary databases.

  Formats of binary logs:

  - Statement: SQL statement-based replication. Binary logs in this format record the SQL statements that are executed to modify data entries.
  - Row: row-based replication. Binary logs in this format record only modification details about data entries in rows.
  - Mixed: replication in mixed mode. This mode combines the statement and row formats. In most cases, binary logs in the statement format are used to record the SQL statements that are executed to modify data entries, such as functions. If data replication from the primary database to secondary databases cannot be implemented by using binary logs in this format, switch to the row format. MySQL determines which format to use based on each SQL statement that is executed.

Limits

Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `v5.x` or `v8.x` . PolarDB for MySQL is not supported.

Procedure

1. Configure a whitelist for the MySQL database.

Add the CIDR block of the VPC in which the exclusive resource group resides to the whitelist of the MySQL database.

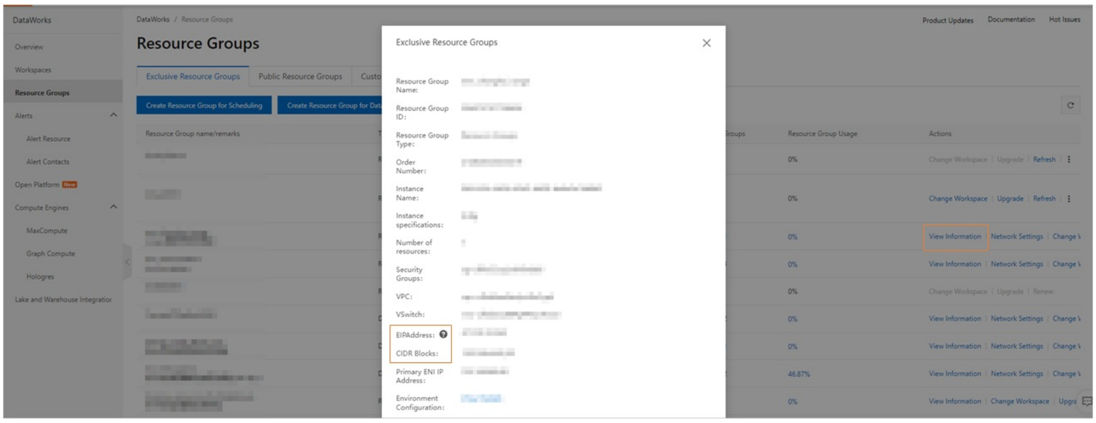
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

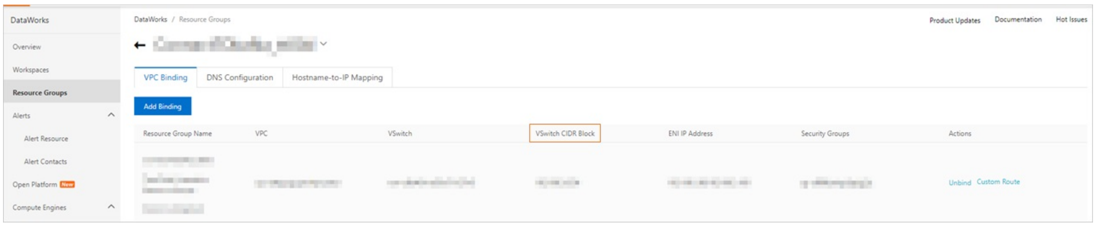
c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.



e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR block of the exclusive resource group to the whitelist of the MySQL database.

2. Create an account and grant permissions to the account.

You must create an account to log on to the MySQL database. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

i. Create an account.

For more information, see Create an account to access a MySQL database.

ii. Grant permissions to the account.

You can run the following command to grant permissions to the account. Alternatively, you can grant the `SUPER` permission to the account. Replace `Account for data synchronization` with the created account when you execute the specific statement.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Password'; // Create an account th
at is used to synchronize data and set a password so that you can use the account and password to acces
s the database from a host. % indicates a host.
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%'; /
/ Grant the SELECT, REPLICATION SLAVE, and REPLICATION CLIENT permissions to the account.
```

`*.*` indicates that the synchronization account is granted the preceding permissions on all tables in all databases. You can also grant the preceding permissions on specified tables in the specified database to the synchronization account. For example, to grant the account the preceding permissions on the *user* table in the *test* database, execute the following statement: `GRANT SELECT, REPLICATION CLIENT ON test.user TO 'Account for data synchronization'@'%';` .

> **Note** The `REPLICATION SLAVE` permission is a global permission. You cannot grant this permission on specified tables in the specified database to the synchronization account.

3. Enable the binary logging feature for the MySQL database.

   Perform the following steps to check whether the binary logging feature is enabled and to query the format of binary logs:

   - Execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_bin";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled.

   - If you use a secondary database to synchronize data, execute the following statement to check whether the binary logging feature is enabled:

     ```
     show variables like "log_slave_updates";
     ```

     If *ON* is displayed in the returned result, the binary logging feature is enabled for the secondary database.

   If the returned result is different from the preceding result, follow the instructions described in the *MySQL documentation* to enable the binary logging feature.

   Execute the following statement to view the format of binary logs:

   ```
   show variables like "binlog_format";
   ```

   Returned result:

   - *ROW*: The format of binary logs is row.
   - *STATEMENT*: The format of binary logs is statement.
   - *MIXED*: The format of binary logs is mixed.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add the source and destination data sources, see Add data sources.

4.8.3. Configure a source Oracle data source

Before you synchronize data from an Oracle data source to Kafka, you can refer to the operations in this topic to configure the network, whitelists, and permissions for the data sources to implement data synchronization.

Prerequisites

Before you configure the data sources, make sure that the following operations are performed:

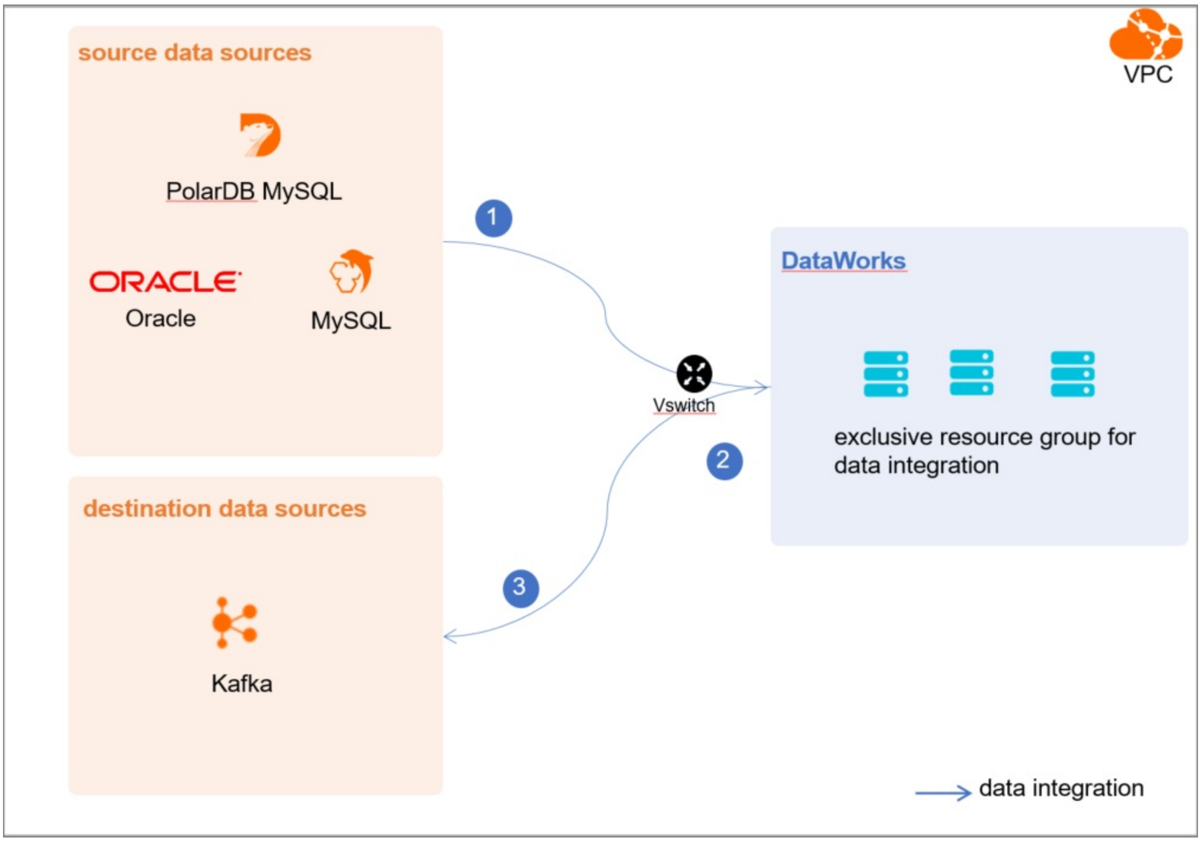
- Prepare data sources: An Oracle data source and a Kafka data source are prepared.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
    - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
    - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and exclusive resource groups for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources. In addition, the version of the Oracle data source must be supported by Data Integration. The Oracle data source must contain only the character encoding formats and data types that are supported by Data Integration.

- Configure whitelists for data sources

  If the source and destination data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Check whether the version of the Oracle data source is available for creating real-time sync nodes of Data Integration in DataWorks

Data synchronization from an Oracle data source in real time is implemented based on the Oracle LogMiner utility that is used to analyze log files. Only the Oracle `10g` , `11g` , `12c non-CDB` , `18c non-CDB` , and `19c non-CDB` databases are supported by real-time sync nodes in DataWorks. The Oracle `12c CDB` , `18c CDB` , and `19c CDB` databases are not supported. An Oracle database of Oracle 12c or later versions can act as a container database (CDB) to host multiple pluggable databases (PDBs).

i. You can execute one of the following statements to view the version of the Oracle database:

- Statement 1:

```
select * from v$version;
```

- Statement 2:

```
select version from v$instance;
```

ii. If the version of the Oracle database is `12c` , `18c` , or `19c` , you can execute the following statement to check whether the database can act as a `CDB` . If the Oracle database can act as a `CDB` , this Oracle database is not supported by real-time sync nodes of Data Integration in DataWorks.

```
select name,cdb,open_mode,con_id from v$database;
```

> ? **Note** You must change the version of the Oracle database that is not supported by real-time sync nodes of Data Integration in DataWorks. Otherwise, you cannot synchronize data from Oracle in real time by using Data Integration.

- Enable the generation of database-level archived log files and redo log files and enable database-level supplemental logging

You must enable the generation of database-level archived log files and redo log files, and database-level supplemental logging for an Oracle data source.

- Archived log files: Oracle archives all redo log files as archived log files that are used to restore the database in the event of a failure.
- Redo log files: Oracle uses redo log files to ensure that database transactions can be re-executed. This way, data can be recovered in the case of a failure such as a power outage.
- Supplemental logging: Supplemental logging is used to supplement the information recorded in redo log files. In Oracle, a redo log file is used to record the values of the fields that are modified. Supplemental logging is used to supplement the change history in the redo log file. This ensures that the redo log file contains complete information that describes data changes. If operations such as data recovery and data synchronization are performed, you can view complete statements and data updates. Some features of the Oracle database can be better implemented after supplemental logging is enabled. Therefore, you must enable supplemental logging for the database.

  For example, if you do not enable supplemental logging, after you execute the UPDATE statement, the redo log file records only the values of the fields that are modified when the UPDATE statement is executed. If you enable supplemental logging, the redo log file records the values of fields before and after a modification. The conditions that are used to modify destination fields are also recorded. When an exception such as a power outage occurs in the database, you can recover data based on the modification details.

  We recommend that you enable supplemental logging for primary key columns or unique index columns.

  - After you enable supplemental logging for primary key columns, the columns that compose a primary key are recorded in logs if the database is updated.
  - After you enable supplemental logging for unique index columns, the columns that compose a unique key or bitmap index are recorded in logs if a column is modified.

Before you synchronize data from an Oracle database in real time by using Data Integration in DataWorks, make sure that the generation of database-level archived log files and supplemental logging are enabled for the Oracle database. To check whether the generation of database-level archived log files and supplementary logging are enabled for the Oracle database, execute the following SQL statement:

```
select log_mode, supplemental_log_data_pk, supplemental_log_data_ui from v$database;
```

- The returned value *ARCHIVELOG* of the `log_mode` parameter indicates that the generation of database-level archived log files is enabled for the Oracle database. If the value *ARCHIVELOG* is not returned, you must *enable supplemental logging* for the Oracle database.

    - The returned value *YES* of the `supplemental_log_data_pk` and `supplemental_log_data_ui` parameters indicates that supplemental logging is enabled for the Oracle database. If the returned value is *FALSE*, you must *enable the generation of database-level archived log files* for the Oracle database.

- Check character encoding formats

  You must make sure that the Oracle database contains only the character encoding formats that are supported by Data Integration to prevent data synchronization failures. The following encoding formats are supported by Data Integration: UTF-8, AL32UTF8, AL16UTF16, and ZHS16GBK.

- Check data types

  You must make sure that the Oracle database contains only the data types that are supported by Data Integration to prevent data synchronization failures. The following data types are not supported by Data Integration for real-time data synchronization: LONG, BFILE, LONG RAW, and NCLOB.

Limits

- You can configure the supplemental logging feature only in a primary Oracle database. Supplemental logging can be enabled for a primary or secondary database.
- The following encoding formats are supported by Data Integration: UTF-8, AL32UTF8, AL16UTF16, and ZHS16GBK.
- The following data types are not supported by Data Integration for real-time data synchronization: LONG, BFILE, LONG RAW, and NCLOB.
- Only the Oracle `10g`, `11g`, `12c non-CDB`, `18c non-CDB`, and `19c non-CDB` databases are supported by real-time sync nodes in DataWorks. The Oracle `12c CDB`, `18c CDB`, and `19c CDB` databases are not supported. An Oracle database of Oracle 12c or later versions can act as a CDB to host multiple PDBs.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add the source and destination data sources, see Add data sources.

4.8.4. Configure a source PolarDB data source

Before you synchronize data from a PolarDB data source to Kafka, you can refer to the operations in this topic to configure the network, whitelists, and permissions for the data sources to implement data synchronization.

Prerequisites

Before you configure the data sources, make sure that the following operations are performed:

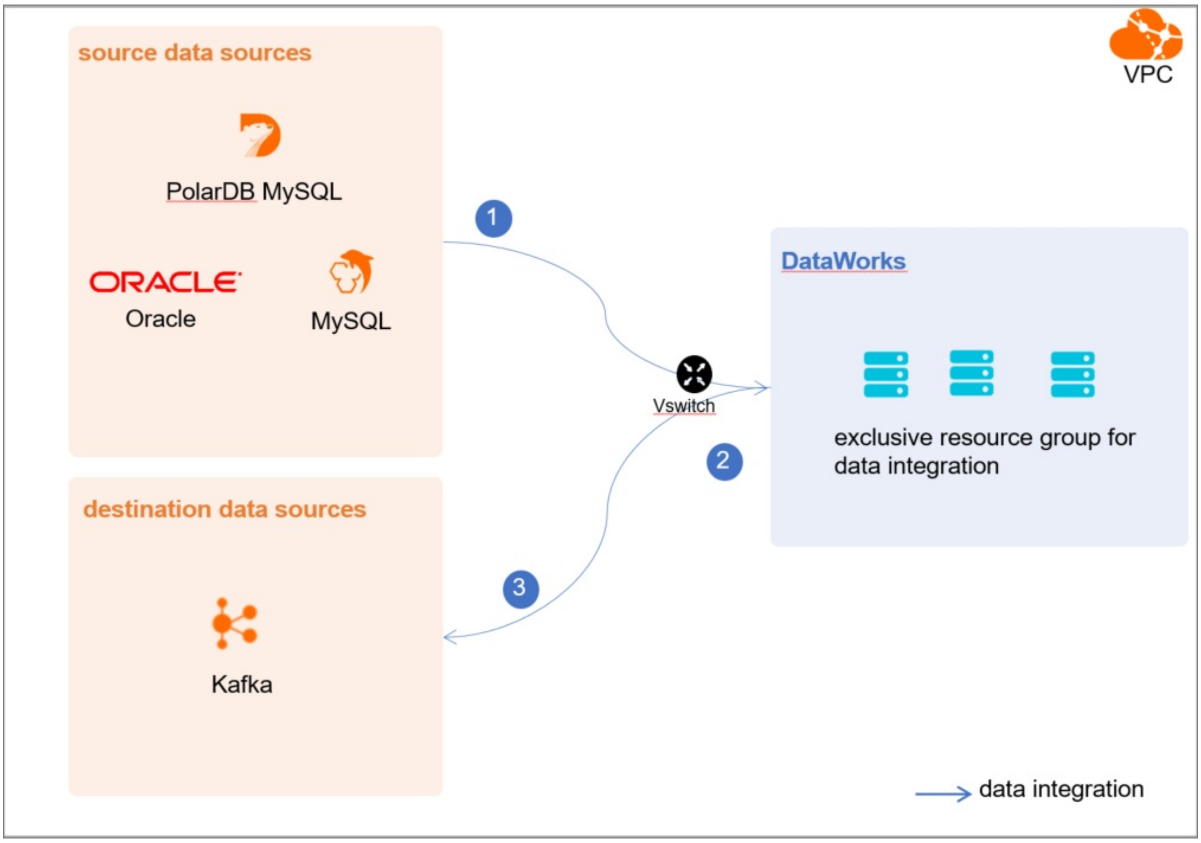
- Prepare data sources: A PolarDB cluster and a Kafka data source are prepared. In this topic, a PolarDB for MySQL cluster is used as the source data source.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you perform data integration, you must select a network connection method based on your business requirements and use the method to connect the data sources to the exclusive resource group for Data Integration. After the data sources and the exclusive resource group for Data Integration are connected, you can refer to the operations described in this topic to configure access settings such as vSwitches and whitelists.
    - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
    - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

Context

Before you synchronize data from the source to the destination, make sure that the data sources and exclusive resource groups for Data Integration are connected. In addition, you must create an account and authorize the account to access the data sources.

- Configure whitelists for data sources

If the source and destination data sources and the exclusive resource group for Data Integration reside in the same VPC, you must add the CIDR block of the exclusive resource group for Data Integration to the whitelists of the data sources. This ensures that the exclusive resource group for Data Integration can be used to access the data sources.



- Create an account and grant permissions the account

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

- Enable the binary logging feature

  If the source is a PolarDB for MySQL cluster, you must enable the binary logging feature for the cluster. PolarDB for MySQL is fully compatible with MySQL and uses high-level physical logs to replace binary logs. To facilitate the integration between PolarDB and the MySQL ecosystem, you can enable the binary logging feature for PolarDB clusters.

Limits

- Only PolarDB for MySQL clusters can be used as sources in data synchronization solutions. Other types of PolarDB data sources are not supported. In this topic, PolarDB indicates PolarDB for MySQL data sources.
- Only data stored on the primary node of a PolarDB for MySQL cluster can be synchronized.

Procedure

1. Configure a whitelist for the PolarDB for MySQL cluster.

   To add the CIDR block of the VPC where the exclusive resource group for Data Integration resides to a whitelist of the PolarDB for MySQL cluster, perform the following steps:

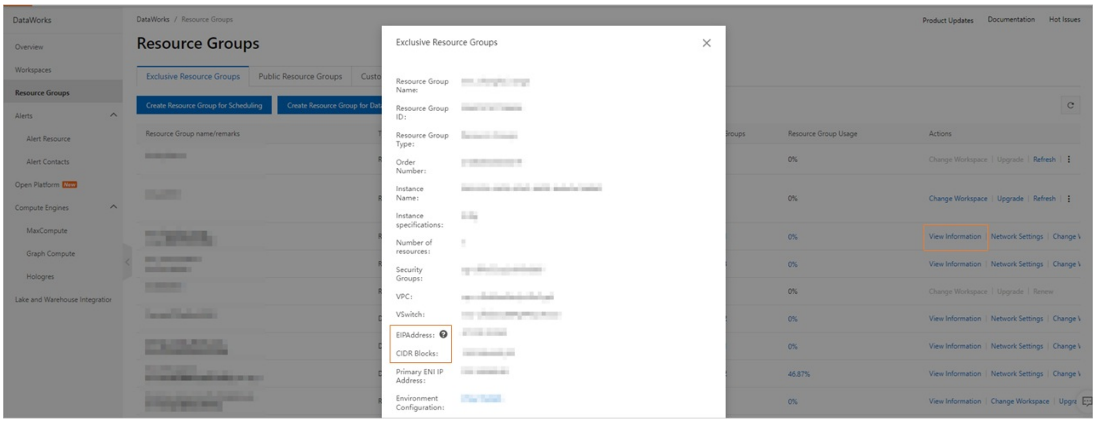
i. View and record the elastic IP address (EIP) and CIDR block of the exclusive resource group for Data Integration.

a. Log on to the DataWorks console.

b. In the left-side navigation pane, click **Resource Groups**.

c. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **View Information** in the Actions column.

d. In the Exclusive Resource Groups dialog box, view and record the values of the **EIPAddress** and **CIDR Blocks** parameters.

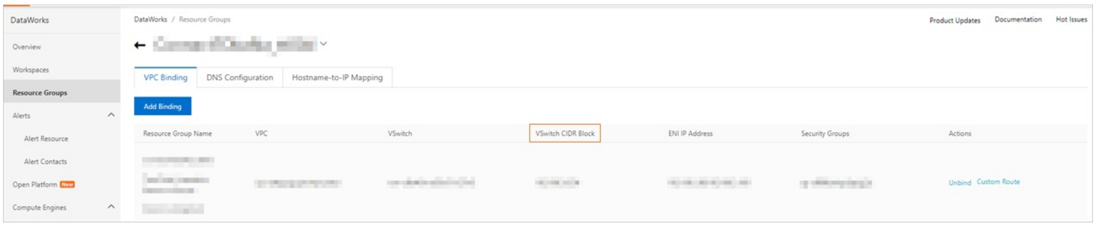

DataWorks
Overview
Workspaces
Resource Groups
Alerts
Alert Resource
Alert Contacts
Open Platform New
Compute Engines
MaxCompute
Graph Compute
Hologres
Lake and Warehouse Integration
DataWorks / Resource Groups
Resource Groups
Exclusive Resource Groups
Public Resource Groups
Create Resource Group for Scheduling
Create Resource Group for Data
Resource Group name/remarks
Resource Group Usage
Actions
Product Updates
Documentation
Hot Issues
View Information | Network Settings | Change
Change Workspace | Upgrade | Refresh

Exclusive Resource Groups

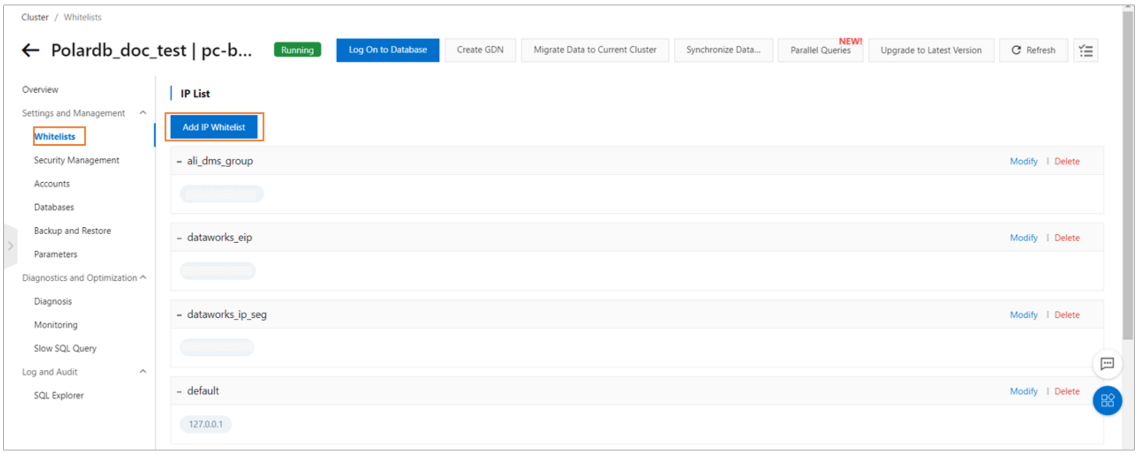
Resource Group Name: [redacted]
Resource Group ID: [redacted]
Resource Group Type: [redacted]
Order Number: [redacted]
Instance Name: [redacted]
Instance specifications: [redacted]
Number of resources: [redacted]
Security Groups: [redacted]
VPC: [redacted]
VSwitch: [redacted]
EIPAddress: [redacted]
CIDR Blocks: [redacted]
Primary ENI IP Address: [redacted]
Environment Configuration: [redacted]

e. On the **Exclusive Resource Groups** tab, find the exclusive resource group for Data Integration and click **Network Settings** in the Actions column.

f. On the **VPC Binding** tab of the page that appears, view and record the **CIDR block of the vSwitch** with which the exclusive resource group for Data Integration is associated.



ii. Add the EIP and CIDR blocks recorded in the preceding steps to the whitelist of the PolarDB for MySQL cluster.



For more information, see Configure an IP whitelist.

2. Create an account and grant the required permissions to the account.

You must create an account to log on to the database of the PolarDB for MySQL cluster. You must grant the `SELECT, REPLICATION SLAVE, and REPLICATION CLIENT` permissions to the account.

i. Create an account.

For more information, see Create a database account.

ii. Grant the required permissions to the account.

You can run the following command to grant the required permissions to the account, or you can directly assign the `SUPER` role to the account.

```
-- CREATE USER 'Account for data synchronization'@'%' IDENTIFIED BY 'Account for data synchronization';
GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'Account for data synchronization'@'%';
```

3. Enable the binary logging feature for the PolarDB for MySQL cluster.

   For more information, see Enable binary logging.

What's next

After the data sources are configured, the source, destination, and exclusive resource group for Data Integration are connected. Then, the exclusive resource group for Data Integration can be used to access the data sources. You can add the source and destination to DataWorks, and associate them with a data synchronization solution when you create the solution.

For more information about how to add the source and destination data sources, see Add data sources.

4.8.5. Add data sources

Before you configure a real-time sync node to synchronize data from a data source to the Kafka data source, you must add both data sources to DataWorks for subsequent source and destination configurations.

Prerequisites

Before you add data sources, make sure that the following operations are performed:

- Prepare data sources: A source data source and a destination data source are created.
- Create and authorize an account: An account that is used to access data sources is created.

Precautions

DataWorks provides workspaces in basic mode and standard mode. A workspace in basic mode does not isolate the development environment from the production environment. A workspace in standard mode isolates the development environment from the production environment.

If you use a workspace in standard mode, you must separately add data sources to the development environment and production environment.

Add a source MySQL data source

To add a source MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source MySQL data source, see Add a MySQL data source.

Add a source Oracle data source

To add a source Oracle data source, you must configure information such as the network connection type, the account, and the password as planned. For more information, see Configure data sources for data synchronization from Oracle.

Add a source PolarDB for MySQL data source

To add a source PolarDB for MySQL data source, you must configure information such as the network connection type and the access account and password as planned. For more information about how to add a source PolarDB for MySQL data source, see Add a PolarDB data source.

If the source PolarDB for MySQL data source that you want to add fails the connectivity test, follow the instructions described in What do I do if the PolarDB data source cannot be connected? to handle the exception.

Add a destination Kafka data source

For information about how to add a destination Kafka data source, see Add a Kafka data source.

What's next

After you add data sources, you can create and run a sync solution to synchronize data from the source data source to the destination data source.

For more information, see Create and configure a sync solution.

4.8.6. Create and configure a sync solution

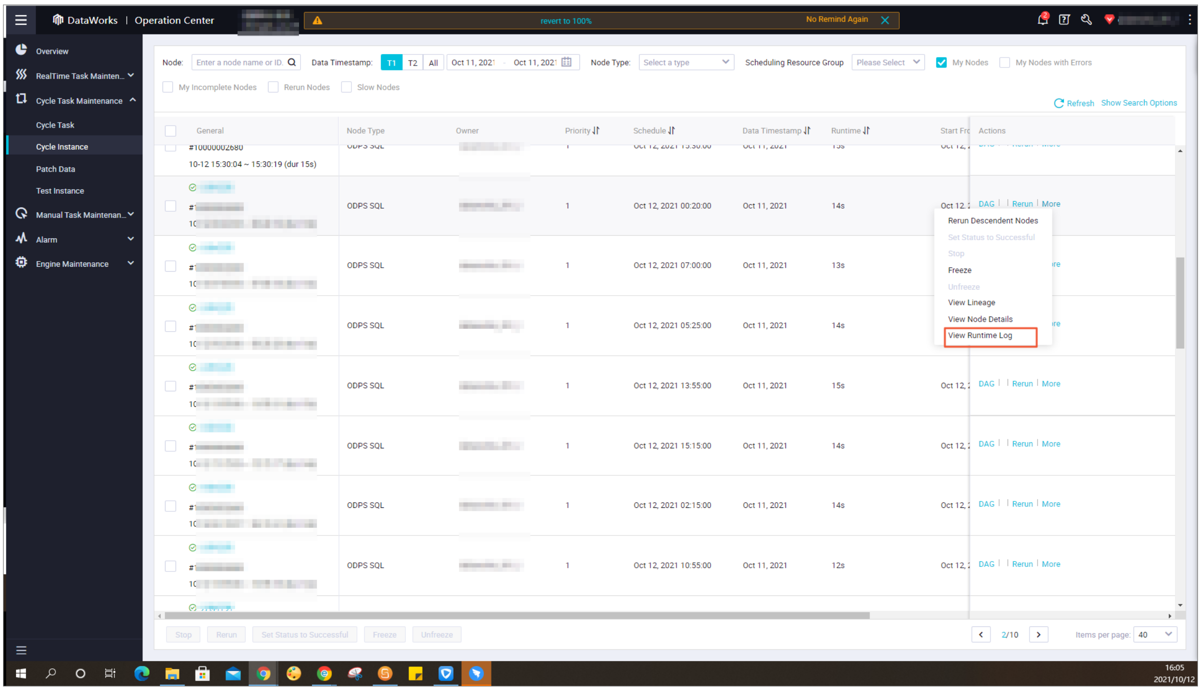
After you configure data sources, network environments, and resource groups, you can create and run a sync solution. This topic describes how to create a sync solution and view the status of the nodes that are generated by the sync solution.

Prerequisites

Before you create a sync solution, make sure that the following operations are performed:

- Plan and configure resources
- Configure a source MySQL data source
- Configure a source Oracle data source
- Configure a source PolarDB data source
- Add data sources
- To run sync nodes on an exclusive resource group for Data Integration, make sure that the version of the **DataX** plug-in that is used to run batch sync nodes is 20210726203000 or later and the version of the **StreamX** plug-in that is used to run real-time sync nodes is 202107121400 or later. Otherwise, you may receive a data format error or fail to run sync nodes for synchronizing incremental data or full data to Kafka.

  View the version of DataX: Go to the **Operation Center** page and click **Patch Data** under Cycle Task Maintenance in the left-side navigation pane. Right-click the batch sync node and click View Runtime Log in the directed acyclic graph (DAG) of the node. On the page that appears, search `Detail log url` in the log area and click the link to go to the page that displays the details of the batch sync node. Then, search for the version information on the page that appears in the format of `DataX( ..... ),From Alibaba!` .For example, search `DataX (20210709_keyindex-20210709144909), From Alibaba !"` in the log area to view the version information of **DataX**, as shown in the third figure in this section.



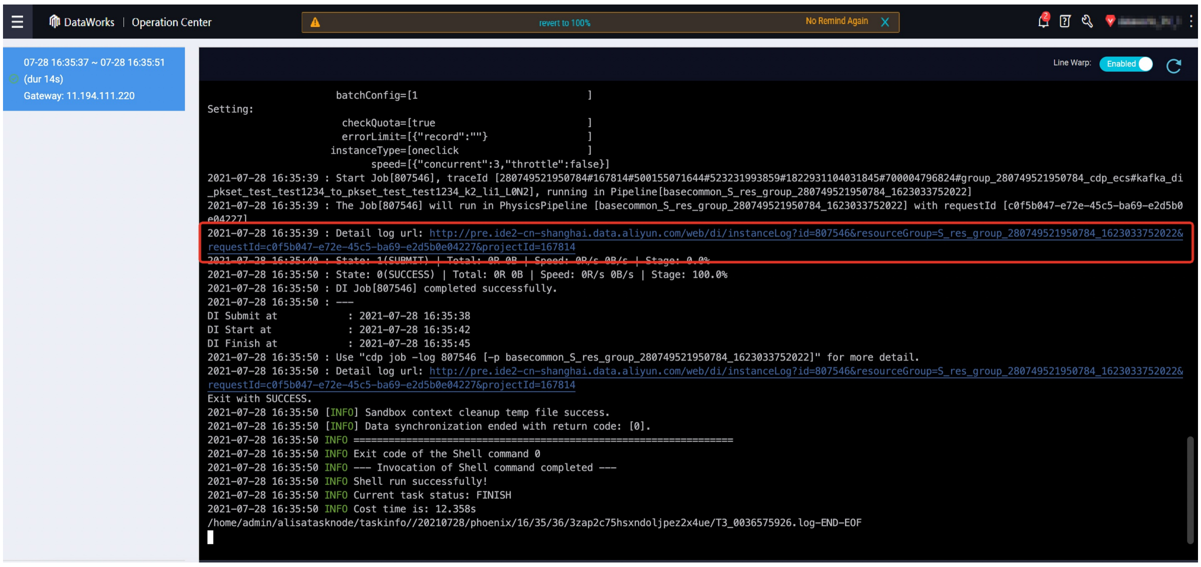
Node: Enter a node name or ID. Data Timestamp: T1 T2 All Oct 11, 2021 - Oct 11, 2021 Node Type: Select a type Scheduling Resource Group Please Select ☑ My Nodes □ My Nodes with Errors

□ My Incomplete Nodes □ Rerun Nodes □ Slow Nodes

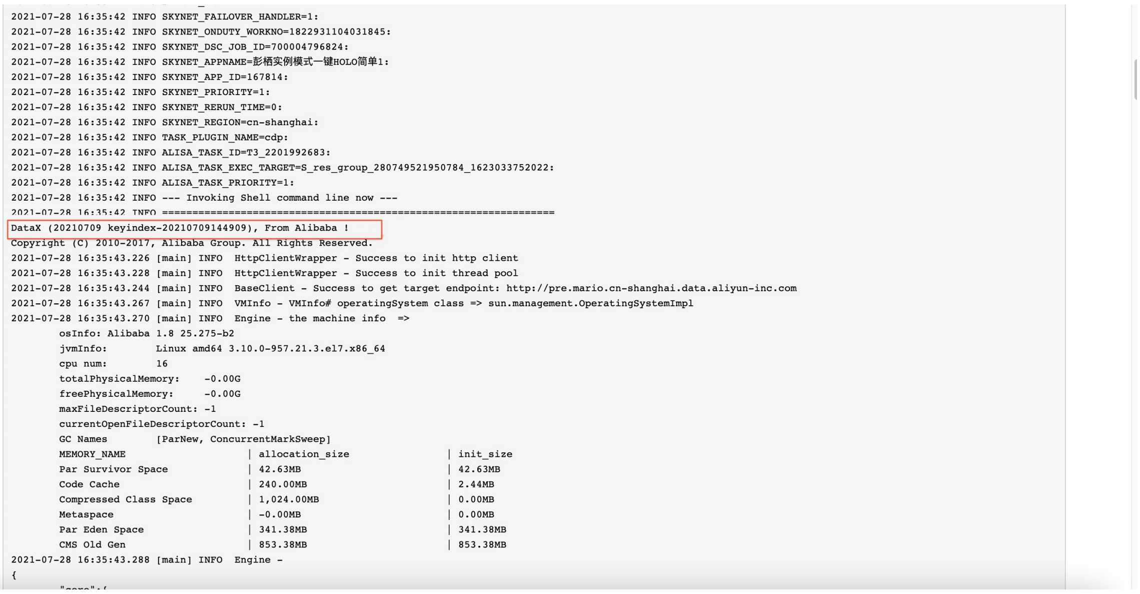
Refresh Show Search Options

General	Node Type	Owner	Priority	Schedule	Data Timestamp	Runtime	Start Fro	Actions
#10000002680 10-12 15:30:04 ~ 15:30:19 (dur 15s)	ODPS SQL	[redacted]	1	Oct 12, 2021 15:30:00	Oct 11, 2021	15s	Oct 12, 2	DAG | Rerun | More
[redacted]	ODPS SQL	[redacted]	1	Oct 12, 2021 00:20:00	Oct 11, 2021	14s	Oct 12, 2	DAG | Rerun | More
[redacted]	ODPS SQL	[redacted]	1	Oct 12, 2021 07:00:00	Oct 11, 2021	13s		re
[redacted]	ODPS SQL	[redacted]	1	Oct 12, 2021 05:25:00	Oct 11, 2021	14s		re
[redacted]	ODPS SQL	[redacted]	1	Oct 12, 2021 13:55:00	Oct 11, 2021	15s	Oct 12, 2	DAG | Rerun | More
[redacted]	ODPS SQL	[redacted]	1	Oct 12, 2021 15:15:00	Oct 11, 2021	14s	Oct 12, 2	DAG | Rerun | More
[redacted]	ODPS SQL	[redacted]	1	Oct 12, 2021 02:15:00	Oct 11, 2021	14s	Oct 12, 2	DAG | Rerun | More
[redacted]	ODPS SQL	[redacted]	1	Oct 12, 2021 10:55:00	Oct 11, 2021	12s	Oct 12, 2	DAG | Rerun | More

2/10 Items per page: 40

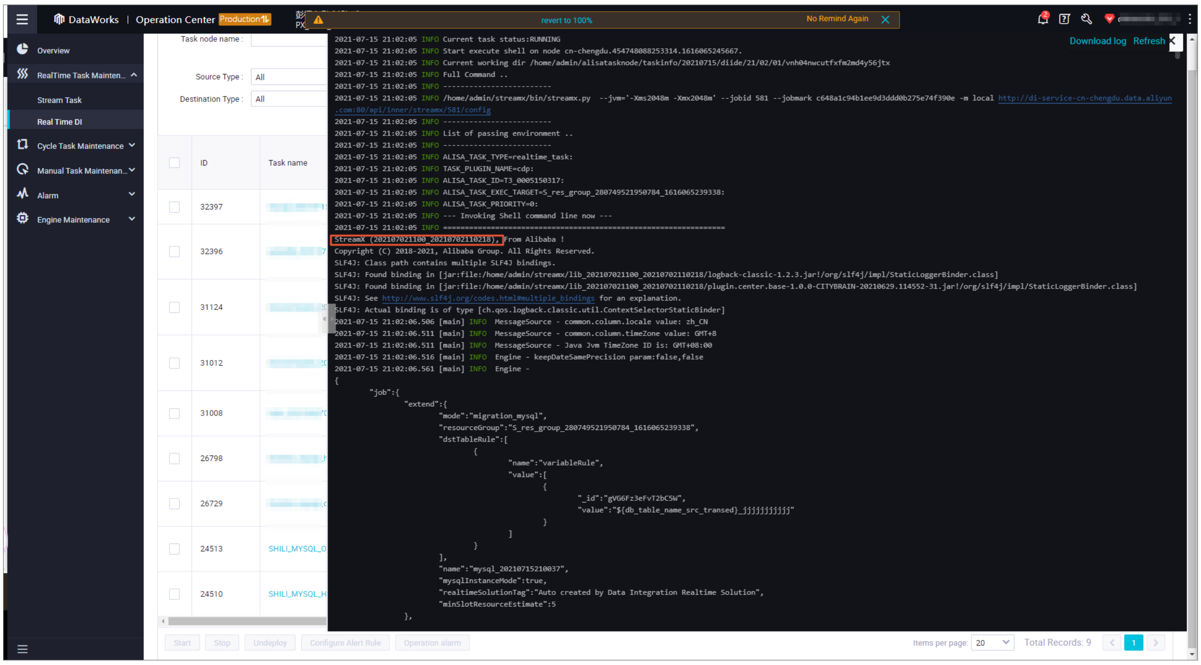


```
                      batchConfig=[1                                          ]
Setting:
                       checkQuota=[true                                       ]
                       errorLimit=[{"record":""}                              ]
                     instanceType=[oneclick                                   ]
                            speed=[{"concurrent":3,"throttle":false}]
2021-07-28 16:35:39 : Start Job[807546], traceId [280749521950784#167814#500155071644#523231993859#1822931104031845#700004796824#group_280749521950784_cdp_ecs#kafka_di_pkset_test_test1234_to_pkset_test_test1234_k2_li1_L0N2], running in Pipeline[basecommon_S_res_group_280749521950784_1623033752022]
2021-07-28 16:35:39 : The Job[807546] will run in PhysicsPipeline [basecommon_S_res_group_280749521950784_1623033752022] with requestId [c0f5b047-e72e-45c5-ba69-e2d5b0e04227]
2021-07-28 16:35:39 : Detail log url: http://pre.ide2-cn-shanghai.data.aliyun.com/web/di/instanceLog?id=807546&resourceGroup=S_res_group_280749521950784_1623033752022&requestId=c0f5b047-e72e-45c5-ba69-e2d5b0e04227&projectId=167814
2021-07-28 16:35:40 : State: 1(SUBMIT) | Total: 0R 0B | Speed: 0R/s 0B/s | Stage: 0.0%
2021-07-28 16:35:50 : State: 0(SUCCESS) | Total: 0R 0B | Speed: 0R/s 0B/s | Stage: 100.0%
2021-07-28 16:35:50 : DI Job[807546] completed successfully.
2021-07-28 16:35:50 : ---
DI Submit at            : 2021-07-28 16:35:38
DI Start at             : 2021-07-28 16:35:42
DI Finish at            : 2021-07-28 16:35:45
2021-07-28 16:35:50 : Use "cdp job -log 807546 [-p basecommon_S_res_group_280749521950784_1623033752022]" for more detail.
2021-07-28 16:35:50 : Detail log url: http://pre.ide2-cn-shanghai.data.aliyun.com/web/di/instanceLog?id=807546&resourceGroup=S_res_group_280749521950784_1623033752022&requestId=c0f5b047-e72e-45c5-ba69-e2d5b0e04227&projectId=167814
Exit with SUCCESS.
2021-07-28 16:35:50 [INFO] Sandbox context cleanup temp file success.
2021-07-28 16:35:50 [INFO] Data synchronization ended with return code: [0].
2021-07-28 16:35:50 INFO =================================================================
2021-07-28 16:35:50 INFO Exit code of the Shell command 0
2021-07-28 16:35:50 INFO --- Invocation of Shell command completed ---
2021-07-28 16:35:50 INFO Shell run successfully!
2021-07-28 16:35:50 INFO Current task status: FINISH
2021-07-28 16:35:50 INFO Cost time is: 12.358s
/home/admin/alisatasknode/taskinfo//20210728/phoenix/16/35/36/3zap2c75hsxndoljpez2x4ue/T3_0036575926.log-END-EOF
```



```
2021-07-28 16:35:42 INFO SKYNET_FAILOVER_HANDLER=1:
2021-07-28 16:35:42 INFO SKYNET_ONDUTY_WORKNO=1822931104031845:
2021-07-28 16:35:42 INFO SKYNET_DSC_JOB_ID=700004796824:
2021-07-28 16:35:42 INFO SKYNET_APPNAME=彭栖实例模式一键HOLO简单1:
2021-07-28 16:35:42 INFO SKYNET_APP_ID=167814:
2021-07-28 16:35:42 INFO SKYNET_PRIORITY=1:
2021-07-28 16:35:42 INFO SKYNET_RERUN_TIME=0:
2021-07-28 16:35:42 INFO SKYNET_REGION=cn-shanghai:
2021-07-28 16:35:42 INFO TASK_PLUGIN_NAME=cdp:
2021-07-28 16:35:42 INFO ALISA_TASK_ID=T3_2201992683:
2021-07-28 16:35:42 INFO ALISA_TASK_EXEC_TARGET=S_res_group_280749521950784_1623033752022:
2021-07-28 16:35:42 INFO ALISA_TASK_PRIORITY=1:
2021-07-28 16:35:42 INFO --- Invoking Shell command line now ---
2021-07-28 16:35:42 INFO =================================================================
DataX (20210709 keyindex-20210709144909), From Alibaba !
Copyright (C) 2010-2017, Alibaba Group. All Rights Reserved.
2021-07-28 16:35:43.226 [main] INFO  HttpClientWrapper - Success to init http client
2021-07-28 16:35:43.228 [main] INFO  HttpClientWrapper - Success to init thread pool
2021-07-28 16:35:43.244 [main] INFO  BaseClient - Success to get target endpoint: http://pre.mario.cn-shanghai.data.aliyun-inc.com
2021-07-28 16:35:43.267 [main] INFO  VMInfo - VMInfo# operatingSystem class => sun.management.OperatingSystemImpl
2021-07-28 16:35:43.270 [main] INFO  Engine - the machine info  =>
        osInfo: Alibaba 1.8 25.275-b2
        jvmInfo:        Linux amd64 3.10.0-957.21.3.el7.x86_64
        cpu num:        16
        totalPhysicalMemory:    -0.00G
        freePhysicalMemory:     -0.00G
        maxFileDescriptorCount: -1
        currentOpenFileDescriptorCount: -1
        GC Names        [ParNew, ConcurrentMarkSweep]
        MEMORY_NAME                     | allocation_size                | init_size
        Par Survivor Space              | 42.63MB                        | 42.63MB
        Code Cache                      | 240.00MB                       | 2.44MB
        Compressed Class Space          | 1,024.00MB                     | 0.00MB
        Metaspace                       | -0.00MB                        | 0.00MB
        Par Eden Space                  | 341.38MB                       | 341.38MB
        CMS Old Gen                     | 853.38MB                       | 853.38MB
2021-07-28 16:35:43.288 [main] INFO  Engine -
{
        "core":{
```

View the version of StreamX: Go to the **Operation Center** page and click **Real Time DI** under RealTime Task Maintenance in the left-side navigation pane. On the Real Time DI page, click the real-time sync node. Then, click the Log tab and search for the version information in the log area in the format of `StreamX( ... ),From Alibaba!`.For example, search `StreamX (202107290000_20210729121213), From Alibaba !` in the log area to view the version information of **StreamX**, as shown in the following figure.



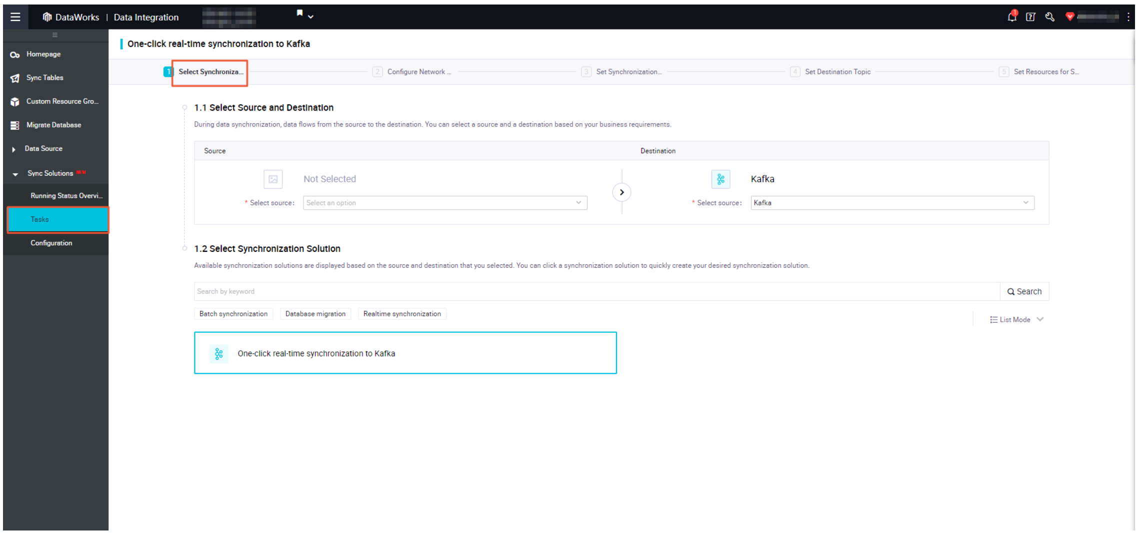
```
2021-07-15 21:02:05 INFO Current task status:RUNNING
2021-07-15 21:02:05 INFO Start execute shell on node cn-chengdu.454748088253314.1616065245667.
2021-07-15 21:02:05 INFO Current working dir /home/admin/alisatasknode/taskinfo/20210715/diide/21/02/01/vnh04nwcutfxfm2md4y56jtx
2021-07-15 21:02:05 INFO Full Command ..
2021-07-15 21:02:05 INFO ------------------------
2021-07-15 21:02:05 INFO /home/admin/streamx/bin/streamx.py  --jvm='-Xms2048m -Xmx2048m' --jobid 581 --jobmark c648a1c94b1ee9d3ddd0b275e74f390e -m local http://di-service-cn-chengdu.data.aliyun
.com:80/api/inner/streamx/581/config
2021-07-15 21:02:05 INFO ------------------------
2021-07-15 21:02:05 INFO List of passing environment ..
2021-07-15 21:02:05 INFO ------------------------
2021-07-15 21:02:05 INFO ALISA_TASK_TYPE=realtime_task:
2021-07-15 21:02:05 INFO TASK_PLUGIN_NAME=cdp:
2021-07-15 21:02:05 INFO ALISA_TASK_ID=T3_0005150317:
2021-07-15 21:02:05 INFO ALISA_TASK_EXEC_TARGET=S_res_group_280749521950784_1616065239338:
2021-07-15 21:02:05 INFO ALISA_TASK_PRIORITY=0:
2021-07-15 21:02:05 INFO --- Invoking Shell command line now ---
2021-07-15 21:02:05 INFO ================================================================
StreamX (202107021100_20210702110218), From Alibaba !
Copyright (C) 2018-2021, Alibaba Group. All Rights Reserved.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/admin/streamx/lib_202107021100_20210702110218/logback-classic-1.2.3.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/admin/streamx/lib_202107021100_20210702110218/plugin.center.base-1.0.0-CITYBRAIN-20210629.114552-31.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [ch.qos.logback.classic.util.ContextSelectorStaticBinder]
2021-07-15 21:02:06.506 [main] INFO  MessageSource - common.column.locale value: zh_CN
2021-07-15 21:02:06.511 [main] INFO  MessageSource - common.column.timeZone value: GMT+8
2021-07-15 21:02:06.511 [main] INFO  MessageSource - Java Jvm TimeZone ID is: GMT+08:00
2021-07-15 21:02:06.516 [main] INFO  Engine - keepDateSamePrecision param:false,false
2021-07-15 21:02:06.561 [main] INFO  Engine -
{
        "job":{
                "extend":{
                        "mode":"migration_mysql",
                        "resourceGroup":"S_res_group_280749521950784_1616065239338",
                        "dstTableRule":[
                                {
                                        "name":"variableRule",
                                        "value":[
                                                {
                                                        "_id":"gVG6Fz3eFvT2bC5W",
                                                        "value":"${db_table_name_src_transed}_jjjjjjjjjj"
                                                }
                                        ]
                                }
                        ],
                        "name":"mysql_20210715210037",
                        "mysqlInstanceMode":true,
                        "realtimeSolutionTag":"Auto created by Data Integration Realtime Solution",
                        "minSlotResourceEstimate":5
                },
```

Create a sync solution

1.
2.
3. Configure a sync solution.
   i. Specify a source data source and a destination data source.

      Select data sources from the **Select source** drop-down lists in the **Source** and **Destination** sections.

      > **Note** The One-click real-time synchronization to Kafka sync solution allows you to synchronize data to Kafka only from MySQL, Oracle, and PolarDB data sources.



1.1 Select Source and Destination

During data synchronization, data flows from the source to the destination. You can select a source and a destination based on your business requirements.

* Select source: Select an option

* Select source: Kafka

1.2 Select Synchronization Solution

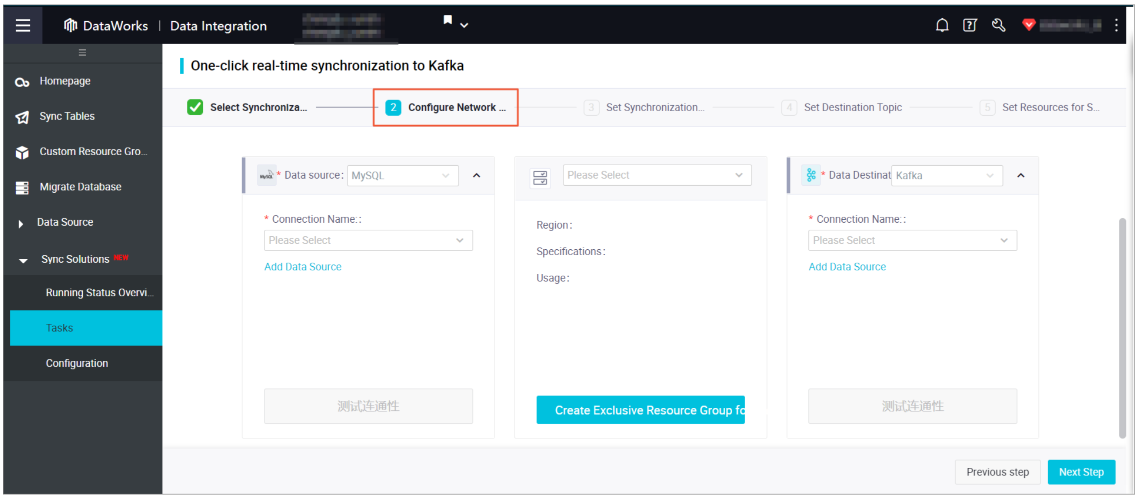
Available synchronization solutions are displayed based on the source and destination that you selected. You can click a synchronization solution to quickly create your desired synchronization solution.

Search by keyword

   ii. In the **Select Synchronization Solution** section, click **One-click real-time synchronization to Kafka**.
   iii. Click **Next Step**.

4. Configure the network connections for data synchronization.

   i. Select data sources from the **Connection Name** drop-down lists for the source and the destination. If no data source is available in the drop-down list, click **Add Data Source** to add one. For more information, see Add a MySQL data source, Add an Oracle data source, and Add a PolarDB data source.

   ii. In the resource group section, select an exclusive resource group for Data Integration from the drop-down list. If no exclusive resource group for Data Integration is available in the drop-down list, click **Create Exclusive Resource Group for Data Integration** to purchase one. In the dialog box that appears, set the **Specifications**, **Number of resources**, and **Billing cycle** parameters and click **Confirm purchase** to go to the payment page. For more information, see Plan and configure resources.

   > Note
   >
   > By default, the Regions parameter is configured based on the region where the workspace resides.
   >
   > After you purchase an exclusive resource group for Data Integration, it is associated with this workspace by default.

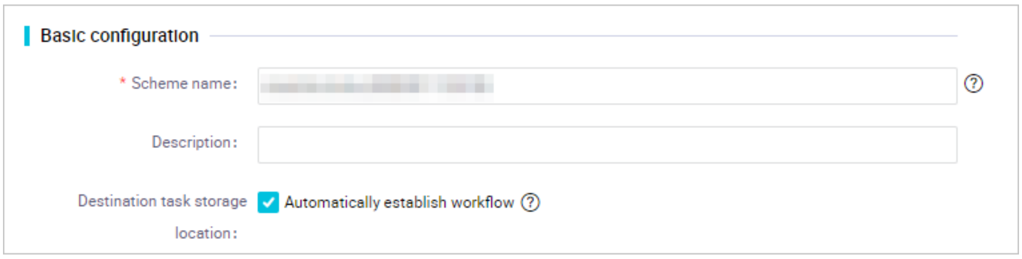
   iii. Click **Test Connectivity** to check the network connections between the exclusive resource group for Data Integration and the data sources. For more information, see Select a network connectivity solution. If the network connectivity test fails, find the cause by following the operations on the **Network Connectivity Diagnostic Tool** tab.

   iv. Click **Next Step**.

5. Select a source and configure synchronization rules.

i. In the Set Synchronization Sources and Rules step, configure basic information such as the solution name for the sync solution.

In the **Basic Configuration** section, set the parameters.



Basic configuration

* Scheme name:

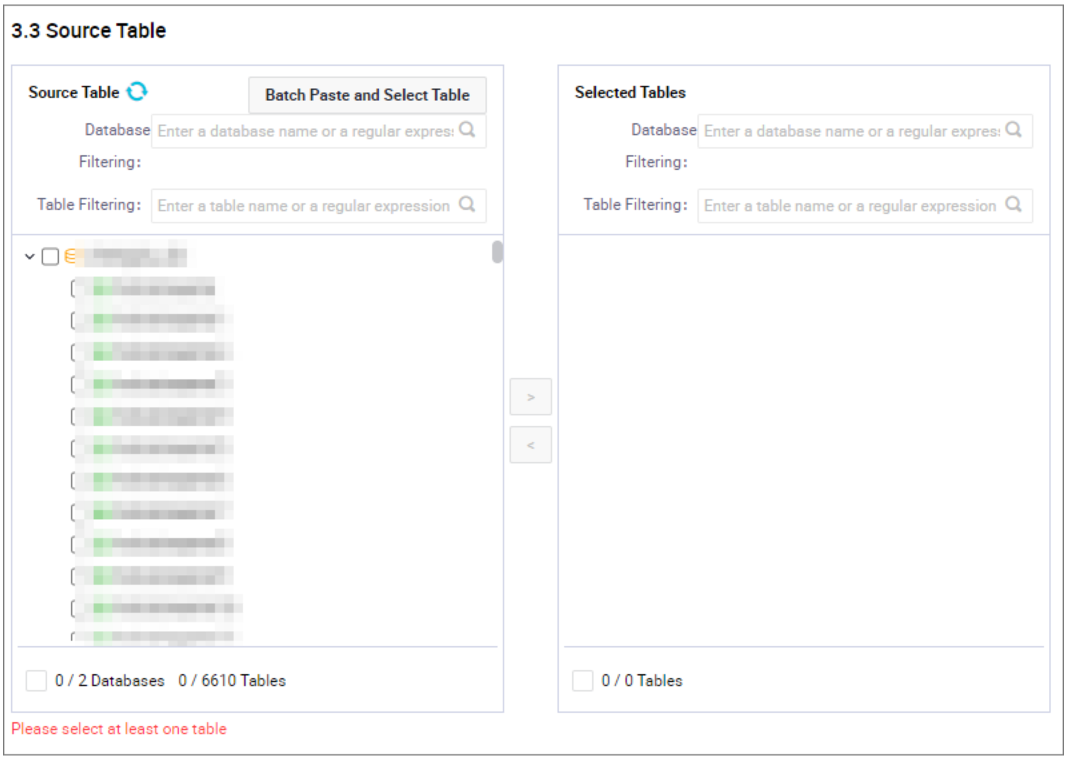
Description:

Destination task storage location: ☑ Automatically establish workflow

Parameter	Description
Solution Name	The name of the sync solution. The name can be up to 50 characters in length.
Description	The description of the sync solution. The description can be up to 50 characters in length.
Location	If you select Automatic Workflow Creation, DataWorks automatically creates a workflow named in the format of clone_database_Source name+to+Destination name. All sync nodes generated by the sync solution are placed in the **Data Integration** folder of this workflow. If you clear **Automatic Workflow Creation**, you must select a directory from the **Select Location** drop-down list. All sync nodes generated by the sync solution are placed in the specified directory.

ii. In the **Data Source** section, select the encoding format for the data source from the **Encoding** drop-down list.

iii. In the **Source Table** section, select the tables whose data you want to synchronize from the **Source Table** list. Then, click the > icon to move the tables to the **Selected Table** list.



3.3 Source Table

Source Table — Batch Paste and Select Table

Database Filtering: Enter a database name or a regular expres

Table Filtering: Enter a table name or a regular expression

□ 0 / 2 Databases 0 / 6610 Tables

Selected Tables

Database Filtering: Enter a database name or a regular expres

Table Filtering: Enter a table name or a regular expression

□ 0 / 0 Tables

Please select at least one table

All tables in the source data source are listed in the Source Table section. You can select all or specified tables to synchronize them at a time.

iv. In the **Mapping Rules for Table Names** section, click **Add rule** to select a rule.

Supported options include **Conversion Rule for Table Name** and **Rule for Destination Table name**.

- **Conversion Rule for Table Name**: the rule that is used to convert the names of source tables into those of destination tables.
- **Rule for Destination Table name**: the rule that is used to add prefixes and suffixes to the names of destination tables.

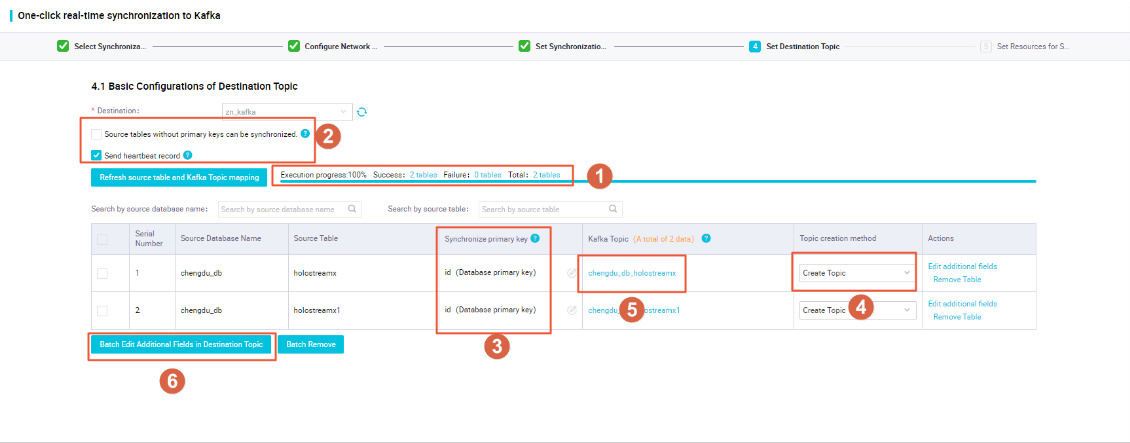
v. Click **Next Step**.

6. Configure the destination topic.

i. By default, the **Destination** parameter is set to the destination data source that you configure.

ii. Click **Refresh source table and Kafka Topic mapping** to configure the mappings between the source tables and destination Kafka topics.

iii. View the mapping progress, source tables, and mapped destination topics.



One-click real-time synchronization to Kafka

☑ Select Synchroniza... ☑ Configure Network ... ☑ Set Synchronizatio... 4 Set Destination Topic 5 Set Resources for S...

4.1 Basic Configurations of Destination Topic

* Destination: zn_kafka

□ Source tables without primary keys can be synchronized.

☑ Send heartbeat record

Refresh source table and Kafka Topic mapping

Execution progress:100% Success: 2 tables Failure: 0 tables Total: 2 tables

Search by source database name: Search by source database name

Search by source table: Search by source table

□	Serial Number	Source Database Name	Source Table	Synchronize primary key	Kafka Topic (A total of 2 data)	Topic creation method	Actions
□	1	chengdu_db	holostreamx	id (Database primary key)	chengdu_db_holostreamx	Create Topic	Edit additional fields Remove Table
□	2	chengdu_db	holostreamx1	id (Database primary key)	chengdu_...ostreamx1	Create Topic	Edit additional fields Remove Table

Batch Edit Additional Fields in Destination Topic

Batch Remove

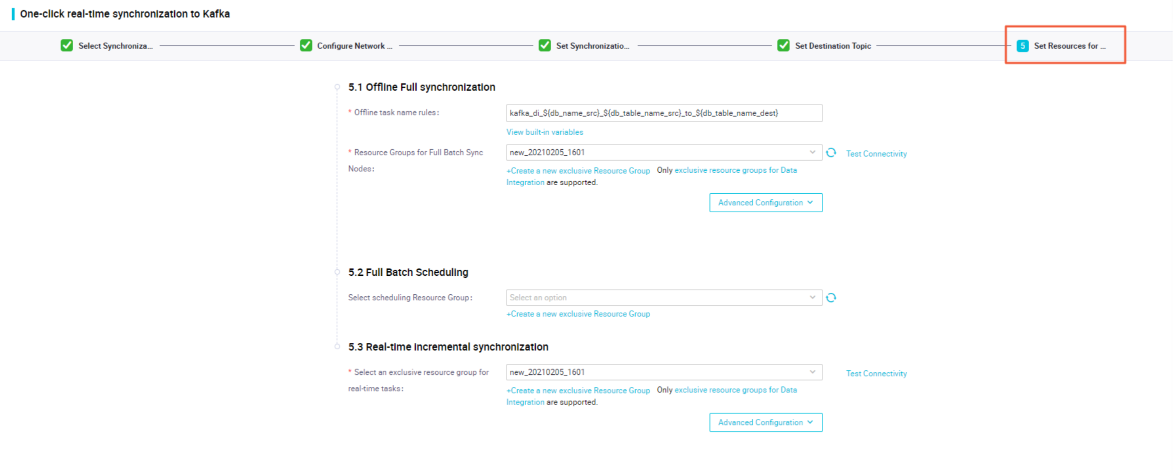
Serial number	Description
1	The progress of mapping the source tables to destination tables. **Note** The mapping may require an extended period of time if you want to synchronize data from a large number of tables.
2	■ If you select **Source tables without primary keys can be synchronized**, a source table that does not contain a primary key can be synchronized to the destination. However, duplicate data may exist if you perform data synchronization. ■ If you select **Send heartbeat record**, the real-time sync node writes a record that contains the current timestamp to Kafka every 5 seconds. This way, you can view the updates of the timestamp for the latest record written to Kafka and check the progress of the data synchronization even if no new records are written to Kafka.
3	■ If the tables in the source database contain primary keys, the system removes duplicate data based on the primary keys during the synchronization. ■ If you select **Source tables without primary keys can be synchronized** and the source table does not contain a primary key, click the ✎ icon to specify a primary key. You can select one or more columns to serve as the primary key. The values of the one or more columns are used to remove duplicate data when you perform data synchronization.
4	The method that is used to create a destination topic. Valid values: **Use Existing Topic** and **Create Topic**.

Serial number	Description
5	The value in the Kafka Topic column varies with the value that you set for **Topic creation method**. ■ If you set the **Topic creation method** parameter to **Use Existing Topic**, you can select the destination topic from the drop-down list in the **Kafka Topic** column. ■ If you set the **Topic creation method** parameter to **Create Topic**, the name of the topic that is automatically created appears in the Kafka Topic column. You can click the automatically created topic to view and modify the name and description of the topic.
6	You can click **Batch Edit Additional Fields in Destination Topic** and add fields for multiple Kafka topics in the dialog box that appears. You can also click **Edit additional fields** in the **Actions** column to add additional fields for a single Kafka topic. **Note** The Batch Edit Additional Fields in Destination Topic feature takes effect only if you select **Create Topic** for the **Topic creation method** parameter.

iv. Click **Next Step**.

7. Configure the resources required by the sync solution.

   In the **Set Resources for Solution Running** step, set the parameters that are described in the following table.



5.1 Offline Full synchronization

* Offline task name rules: kafka_di_${db_name_src}_${db_table_name_src}_to_${db_table_name_dest}

* Resource Groups for Full Batch Sync Nodes: new_20210205_1601

Only exclusive resource groups for Data Integration are supported.

5.2 Full Batch Scheduling

Select scheduling Resource Group: Select an option

5.3 Real-time incremental synchronization

* Select an exclusive resource group for real-time tasks: new_20210205_1601

Only exclusive resource groups for Data Integration are supported.

- **Offline Sync**

Parameter	Description
Offline task name rules	The name of the batch sync node that is used to synchronize the full data of the source. After a sync solution is created, DataWorks first generates a batch sync node to synchronize full data, and then generates real-time sync nodes to synchronize incremental data.
Resource Groups for Full Batch Sync Nodes	The exclusive resource group for Data Integration that is used to run the batch sync node. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- **Scheduling Settings**

Parameter	Description
Select scheduling Resource Group	The resource group for scheduling that is used to run the nodes. Only exclusive resource groups for Data Integration can be used to run sync solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Incremental Sync

Parameter	Description
Resource Groups for Incremental Batch Sync Nodes	The exclusive resource group that is used to run the real-time sync nodes. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Channel Settings

Parameter	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: **20**.

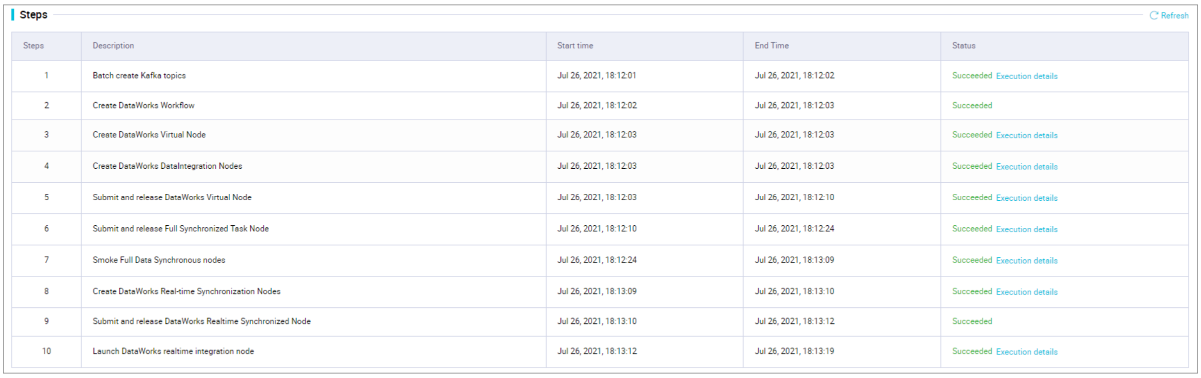
8. Click **Complete Configuration**. The real-time sync solution used to synchronize all data in a database is created.

Run the sync solution

On the **Solution task list** tab of the Tasks page, find the configured sync solution and click **Submit and Run** in the Operation column to run the sync solution.

View the status and result of the sync nodes

- On the **Solution task list** tab of the Tasks page, find the solution that is run and choose **Execution details** in the Operation column. Then, you can view the running details of all nodes.



Steps	Description	Start time	End Time	Status
1	Batch create Kafka topics	Jul 26, 2021, 18:12:01	Jul 26, 2021, 18:12:02	Succeeded Execution details
2	Create DataWorks Workflow	Jul 26, 2021, 18:12:02	Jul 26, 2021, 18:12:03	Succeeded
3	Create DataWorks Virtual Node	Jul 26, 2021, 18:12:03	Jul 26, 2021, 18:12:03	Succeeded Execution details
4	Create DataWorks DataIntegration Nodes	Jul 26, 2021, 18:12:03	Jul 26, 2021, 18:12:03	Succeeded Execution details
5	Submit and release DataWorks Virtual Node	Jul 26, 2021, 18:12:03	Jul 26, 2021, 18:12:10	Succeeded Execution details
6	Submit and release Full Synchronized Task Node	Jul 26, 2021, 18:12:10	Jul 26, 2021, 18:12:24	Succeeded Execution details
7	Smoke Full Data Synchronous nodes	Jul 26, 2021, 18:12:24	Jul 26, 2021, 18:13:09	Succeeded Execution details
8	Create DataWorks Real-time Synchronization Nodes	Jul 26, 2021, 18:13:09	Jul 26, 2021, 18:13:10	Succeeded Execution details
9	Submit and release DataWorks Realtime Synchronized Node	Jul 26, 2021, 18:13:10	Jul 26, 2021, 18:13:12	Succeeded
10	Launch DataWorks realtime integration node	Jul 26, 2021, 18:13:12	Jul 26, 2021, 18:13:19	Succeeded Execution details

- Find a node whose execution details you want to view and click **Execution details** in the Status column. In the dialog box that appears, click the provided link to go to the DataStudio page.

Manage the sync solution

- View or edit the sync solution.

  On the **Solution task list** tab of the Tasks page, find the created sync solution and choose **More > View Configuration** or choose **More > Modify Configuration** in the Operation column. Then, you can view or modify the configurations of the sync solution.

- Delete the real-time sync solution.

  Find the real-time sync solution that you want to delete and choose **More > Delete** in the Operation column. In the **Delete** message, click **OK**.

  > ⓘ **Note** After you click OK, only the configuration record of the real-time sync solution is deleted. The sync nodes generated by the sync solution and data tables generated by the sync nodes are not affected.

- Change the priority for the batch synchronization solution

  Find the newly created batch synchronization solution and choose **More > Change Priority** in the Operation column. In the **Change Priority** dialog box, enter the desired priority and click **Confirm**. You can set the priority to an integer from 1 to 8. A larger value indicates a higher priority.

  > ⓘ **Note** If multiple batch synchronization solutions have the same priority, the system runs them based on the order they are committed.

Set the formats of messages written to Kafka

If you run a real-time sync node after you configure a real-time sync solution, the node reads all the existing data from the source database and writes it to the Kafka topics in JSON format. It also reads incremental data and writes the incremental data to Kafka in real time. Besides, it also synchronizes incremental DDL-based data changes from the source database to Kafka in JSON format in real time. For more information about the formats of messages written to Kafka, see Appendix: Message formats.

> ⓘ **Note** If you run a batch sync node to synchronize data to Kafka, the payload.sequenceId, payload.timestamp.eventTIme, and payload.timestamp.checkpointTime fields are set to -1 in the messages written to Kafka. The messages are in JSON format.

4.8.7. Add or remove source tables to or from a sync solution that is running

After you run a sync solution to synchronize data to Kafka, you can add or remove source tables to or from the sync solution with a few clicks. This topic describes how to add or remove source tables to or from a sync solution that is running.

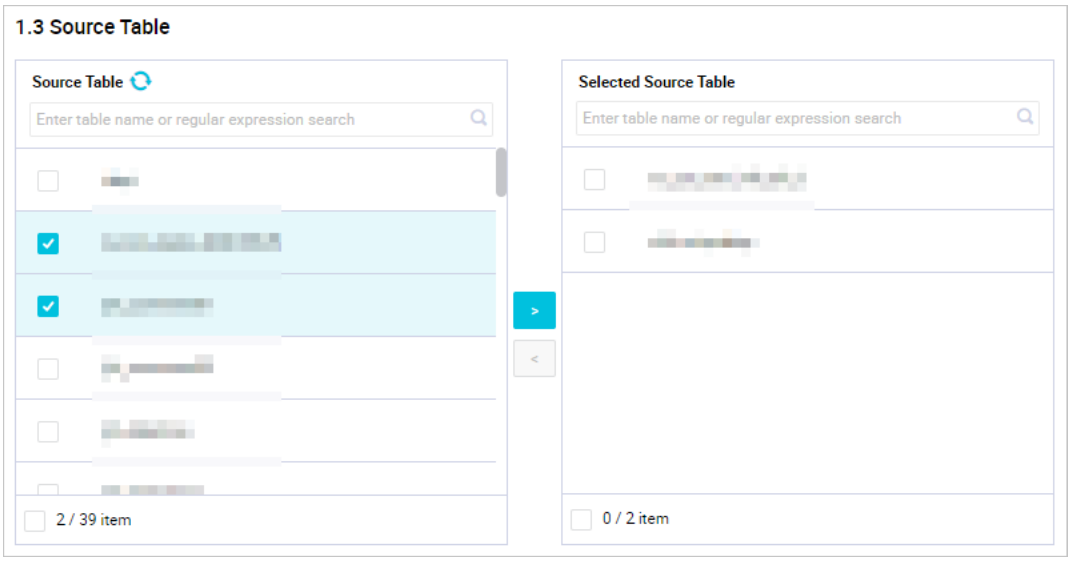
Prerequisites

A sync solution used to synchronize data to Kafka is created and running. For more information, see Create and configure a sync solution.

Add source tables to a sync solution

1.
2. On the **Tasks** tab, find the sync solution and choose **More > Modify Configuration** to go to the solution configuration page.
3. Add source tables to the sync solution and update the mappings between the source tables and destination tables.

i. In the **Source Table** section of the **Set Synchronization Sources and Rules** step, select the source tables that you want to add to the sync solution from the **Source Table** list and click the > icon to move the tables to the **Selected Tables** list.



1.3 Source Table

Source Table

Enter table name or regular expression search

- □ [redacted]
- ☑ [redacted]
- ☑ [redacted]
- □ [redacted]
- □ [redacted]
- □ [redacted]

□ 2 / 39 item

Selected Source Table

Enter table name or regular expression search

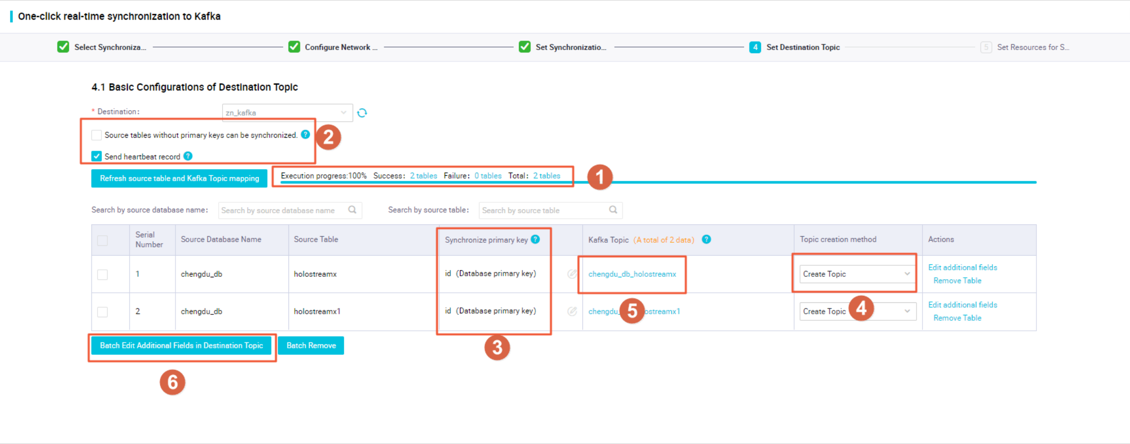
- □ [redacted]
- □ [redacted]

□ 0 / 2 item

ii. Click **Next Step**.

iii. In the **Set Destination Topic** step, click **Refresh source table and Kafka Topic mapping** to configure the mappings between the source tables and destination Kafka topics.

iv. View the mapping progress, source tables, and mapped destination topics.



One-click real-time synchronization to Kafka

4.1 Basic Configurations of Destination Topic

* Destination: zn_kafka

□ Source tables without primary keys can be synchronized.

☑ Send heartbeat record

Refresh source table and Kafka Topic mapping

Execution progress:100% Success: 2 tables Failure: 0 tables Total: 2 tables

Search by source database name: Search by source database name

Search by source table: Search by source table

□	Serial Number	Source Database Name	Source Table	Synchronize primary key		Kafka Topic (A total of 2 data)	Topic creation method	Actions
□	1	chengdu_db	holostreamx	id (Database primary key)		chengdu_db_holostreamx	Create Topic	Edit additional fields Remove Table
□	2	chengdu_db	holostreamx1	id (Database primary key)		chengdu_ ostreamx1	Create Topic	Edit additional fields Remove Table

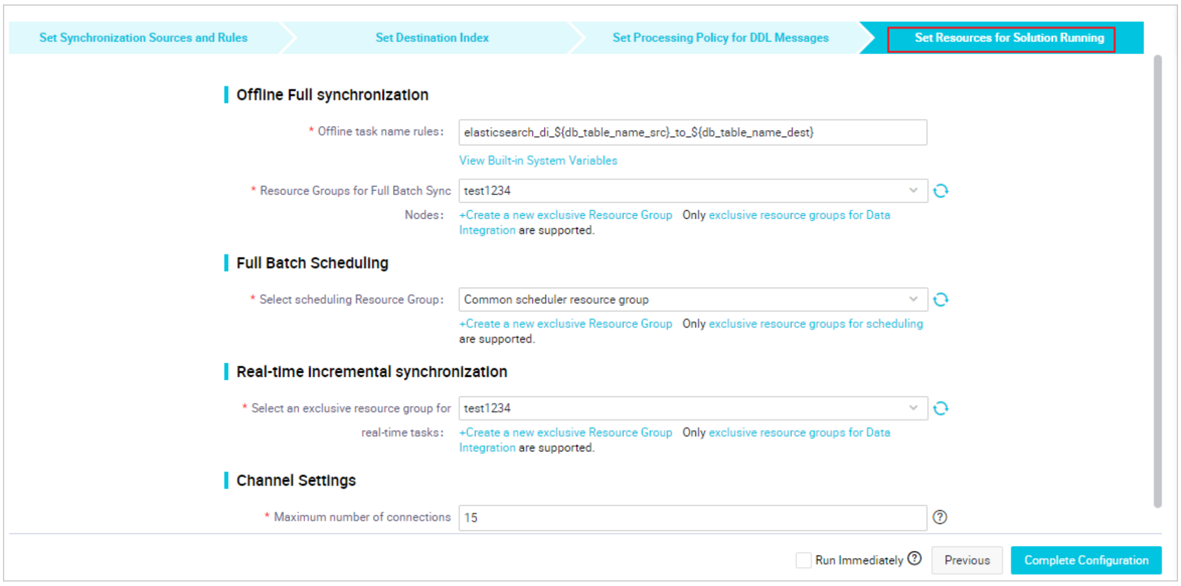
Batch Edit Additional Fields in Destination Topic | Batch Remove

Serial number	Description
1	The progress of mapping the source tables to destination tables. Note The mapping may require an extended period of time if you want to synchronize data from a large number of tables.
2	■ If you select **Source tables without primary keys can be synchronized.**, a source table that does not contain a primary key can be synchronized to the destination. However, duplicate data may exist if you perform data synchronization. ■ If you select **Send heartbeat record**, the real-time sync node writes a record that contains the current timestamp to Kafka every 5 seconds. This way, you can view the updates of the timestamp for the latest record written to Kafka and check the progress of the data synchronization even if no new records are written to Kafka.

Serial number	Description
3	■ If the tables in the source database contain primary keys, the system removes duplicate data based on the primary keys during the synchronization. ■ If you select **Source tables without primary keys can be synchronized.** and the source table does not contain a primary key, click the ✎ icon to specify a primary key. You can select one or more columns to serve as the primary key. The values of the one or more columns are used to remove duplicate data when you perform data synchronization.
4	The method that is used to create a destination topic. Valid values: **Use Existing Topic** and **Create Topic**.
5	The value in the Kafka Topic column varies with the value that you set for **Topic creation method**. ■ If you set the **Topic creation method** parameter to **Use Existing Topic**, you can select the destination topic from the drop-down list in the **Kafka Topic** column. ■ If you set the **Topic creation method** parameter to **Create Topic**, the name of the topic that is automatically created appears in the Kafka Topic column. You can click the automatically created topic to view and modify the name and description of the topic.
6	You can click **Batch Edit Additional Fields in Destination Topic** and add fields for multiple Kafka topics in the dialog box that appears. You can also click **Edit additional fields** in the **Actions** column to add additional fields for a single Kafka topic. **Note** The Batch Edit Additional Fields in Destination Topic feature takes effect only If you select **Create Topic** for the **Topic creation method** parameter.

4. Click **Next Step**.
5. Configure the resources required by the sync solution.

   In the **Set Resources for Solution Running** step, set the parameters as required.


Set Synchronization Sources and Rules
Set Destination Index
Set Processing Policy for DDL Messages
Set Resources for Solution Running

Offline Full synchronization

* Offline task name rules: elasticsearch_di_${db_table_name_src}_to_${db_table_name_dest}

View Built-in System Variables

* Resource Groups for Full Batch Sync Nodes: test1234

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

Full Batch Scheduling

* Select scheduling Resource Group: Common scheduler resource group

+Create a new exclusive Resource Group Only exclusive resource groups for scheduling are supported.

Real-time incremental synchronization

* Select an exclusive resource group for real-time tasks: test1234

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

Channel Settings

* Maximum number of connections: 15

- Offline Sync

Parameter	Description
Offline task name rules	The name of the batch sync node that is used to synchronize the full data of the source. After a sync solution is created, DataWorks first generates a batch sync node to synchronize full data, and then generates real-time sync nodes to synchronize incremental data.

Parameter	Description
Resource Groups for Full Batch Sync Nodes	The exclusive resource group for Data Integration that is used to run the batch sync node. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- **Scheduling Settings**

Parameter	Description
Select scheduling Resource Group	The resource group for scheduling that is used to run the nodes. Only exclusive resource groups for Data Integration can be used to run sync solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- **Incremental Sync**

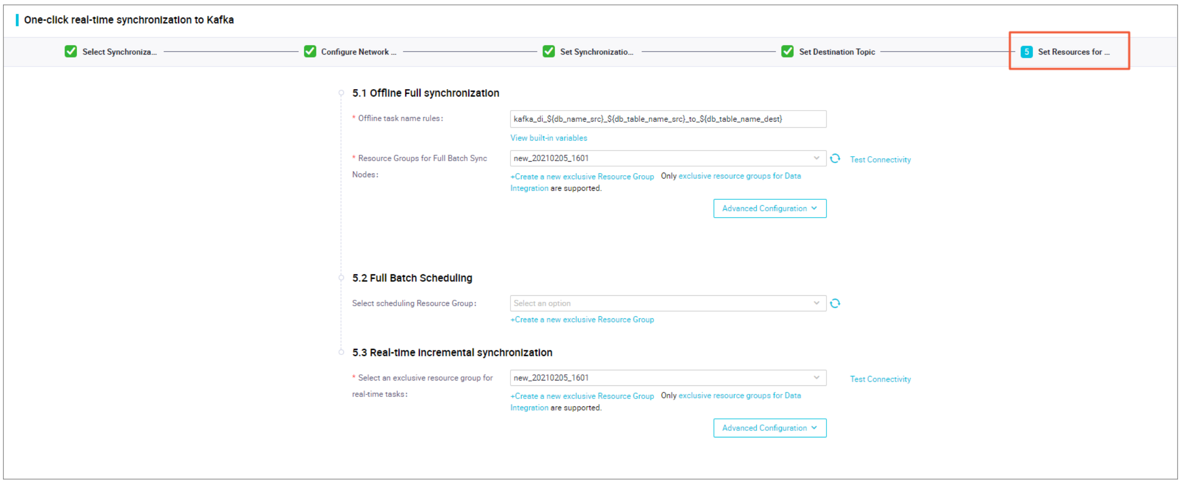
Parameter	Description
Resource Groups for Incremental Batch Sync Nodes	The exclusive resource group that is used to run the real-time sync nodes. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- **Channel Settings**

Parameter	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: **20**.

6. Configure the resources required by the sync solution.

   In the **Set Resources for Solution Running** step, set the parameters that are described in the following table.



One-click real-time synchronization to Kafka

5.1 Offline Full synchronization

* Offline task name rules: kafka_di_${db_name_src}_${db_table_name_src}_to_${db_table_name_dest}

View built-in variables

* Resource Groups for Full Batch Sync Nodes: new_20210205_1601

Only exclusive resource groups for Data Integration are supported.

5.2 Full Batch Scheduling

Select scheduling Resource Group: Select an option

5.3 Real-time incremental synchronization

* Select an exclusive resource group for real-time tasks: new_20210205_1601

Only exclusive resource groups for Data Integration are supported.

- Offline Sync

Parameter	Description
Offline task name rules	The name of the batch sync node that is used to synchronize the full data of the source. After a sync solution is created, DataWorks first generates a batch sync node to synchronize full data, and then generates real-time sync nodes to synchronize incremental data.
Resource Groups for Full Batch Sync Nodes	The exclusive resource group for Data Integration that is used to run the batch sync node. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Scheduling Settings

Parameter	Description
Select scheduling Resource Group	The resource group for scheduling that is used to run the nodes. Only exclusive resource groups for Data Integration can be used to run sync solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Incremental Sync

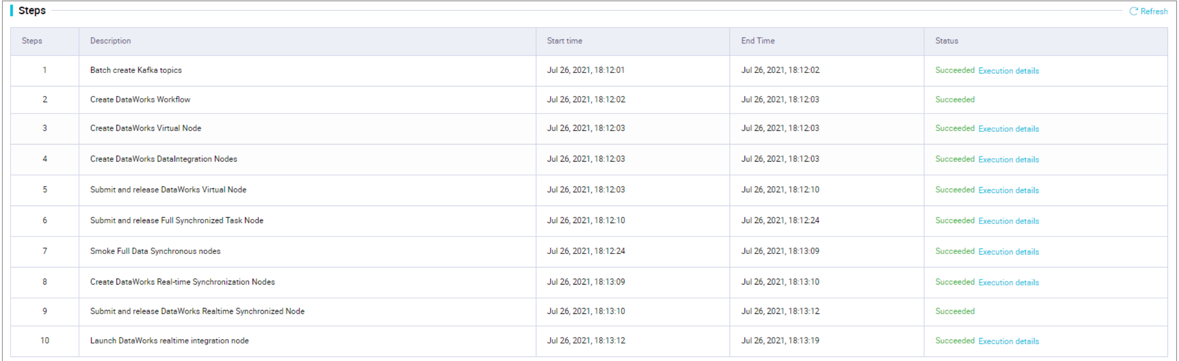
Parameter	Description
Resource Groups for Incremental Batch Sync Nodes	The exclusive resource group that is used to run the real-time sync nodes. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. Note If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- **Channel Settings**

Parameter	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: 20.

7. Click **Complete Configuration** to return to the **Tasks** tab.
8. Find the sync solution to which you added source tables and choose **More > Submit and Run** in the **Operation** column. In the **Submit and Run** message, click **OK** to run the solution.

   After you submit and run the sync solution to which you added source tables, the system compares the source tables in the original sync solution with the source tables in the new sync solution. If new source tables are detected, the system performs the process of adding the source tables.



Steps

Steps	Description	Start time	End Time	Status
1	Batch create Kafka topics	Jul 26, 2021, 18:12:01	Jul 26, 2021, 18:12:02	Succeeded Execution details
2	Create DataWorks Workflow	Jul 26, 2021, 18:12:02	Jul 26, 2021, 18:12:03	Succeeded
3	Create DataWorks Virtual Node	Jul 26, 2021, 18:12:03	Jul 26, 2021, 18:12:03	Succeeded Execution details
4	Create DataWorks DataIntegration Nodes	Jul 26, 2021, 18:12:03	Jul 26, 2021, 18:12:03	Succeeded Execution details
5	Submit and release DataWorks Virtual Node	Jul 26, 2021, 18:12:03	Jul 26, 2021, 18:12:10	Succeeded Execution details
6	Submit and release Full Synchronized Task Node	Jul 26, 2021, 18:12:10	Jul 26, 2021, 18:12:24	Succeeded Execution details
7	Smoke Full Data Synchronous nodes	Jul 26, 2021, 18:12:24	Jul 26, 2021, 18:13:09	Succeeded Execution details
8	Create DataWorks Real-time Synchronization Nodes	Jul 26, 2021, 18:13:09	Jul 26, 2021, 18:13:10	Succeeded Execution details
9	Submit and release DataWorks Realtime Synchronized Node	Jul 26, 2021, 18:13:10	Jul 26, 2021, 18:13:12	Succeeded
10	Launch DataWorks realtime integration node	Jul 26, 2021, 18:13:12	Jul 26, 2021, 18:13:19	Succeeded Execution details

> **Note** After you add source tables to the sync solution at a specific point in time, the system starts to load data to these newly added source tables at this point in time. After the data loading ends, the system starts to synchronize data in these source tables to the destination. For example, your sync solution starts to run at 08:00 and is still running at 09:00. You add a source table to the sync solution at 09:00. Then, the system starts to load data to the table from 09:00, and the loading ends at 10:00. In this case, the system stops the real-time sync nodes that are running and starts to synchronize the data that is generated from 09:00 to 10:00 in the newly added source table to the destination Kafka table. The addition of source tables to a sync solution that is running can ensure only the consistency between data before and after the synchronization.

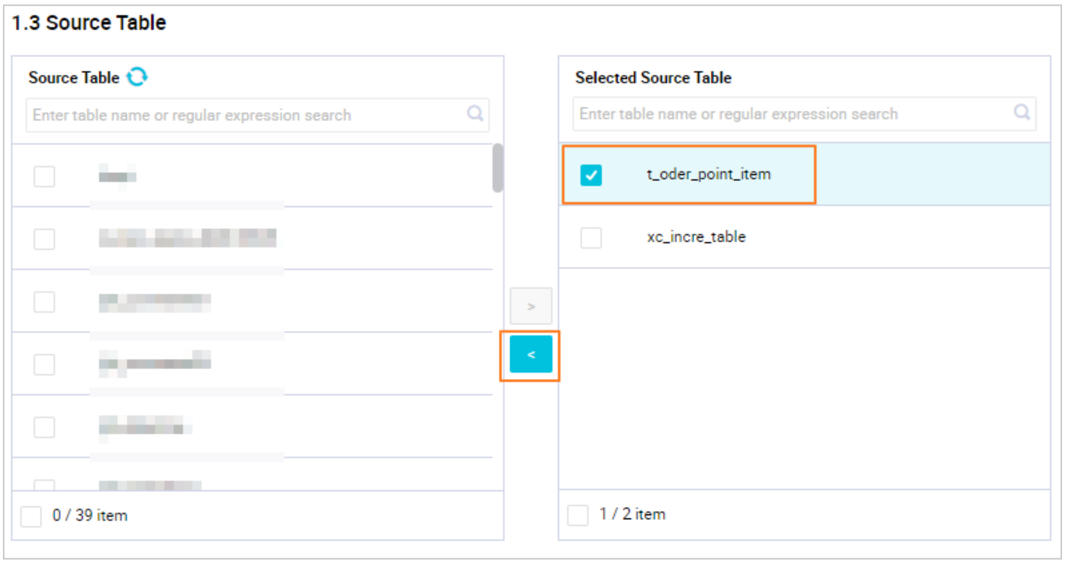
9. View the addition details of the source tables.
   1. On the **Tasks** tab, find the sync solution to which you added source tables and click **Execution details** in the **Operation** column to go to the details page of the sync solution.
   2. In the **steps** section, find the **Display the increased/decreased table** node and click **Execution details** in the Status column.

      If the status of the **Display the increased/decreased table** node is **Succeeded**, the new source tables are added to the sync solution.
   3. View the new source tables that are added to the sync solution.

Remove source tables from the sync solution

1.
2. On the **Tasks** tab, find the sync solution and choose **More > Modify Configuration** to go to the solution configuration page.
3. Remove source tables from the sync solution and update the mappings between the remaining source tables and destination tables.

i. In the **Source Table** section of the **Set Synchronization Sources and Rules** step, select the source tables that you want to remove from the sync solution in the **Selected Tables** list and click the < icon to move the tables back to the **Source Table** list.



1.3 Source Table

Source Table

Enter table name or regular expression search

□ [redacted]

□ [redacted]

□ [redacted]

□ [redacted]

□ [redacted]

□ [redacted]

□ 0 / 39 item

Selected Source Table

Enter table name or regular expression search

☑ t_oder_point_item

□ xc_incre_table

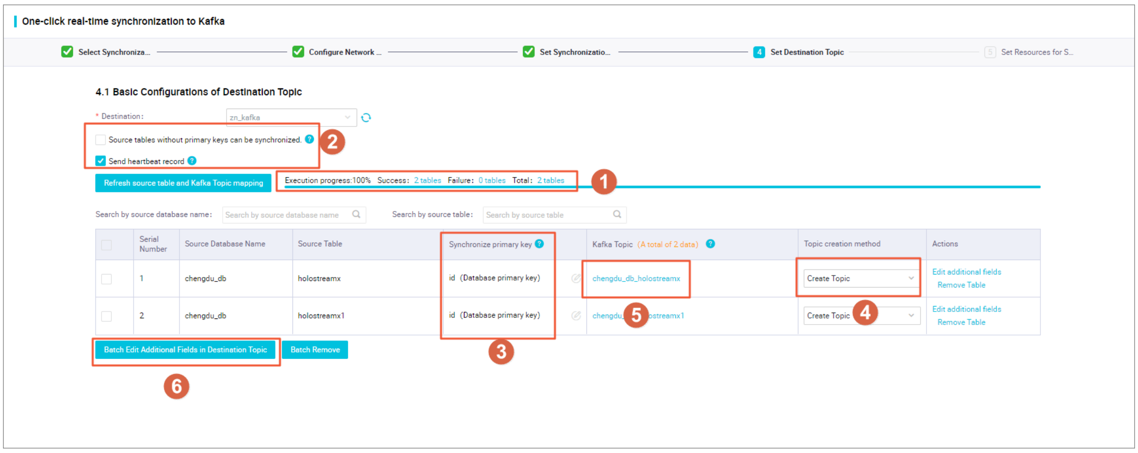
□ 1 / 2 item

ii. Click **Next Step**.

iii. In the **Set Destination Topic** step, click **Refresh source table and Kafka Topic mapping** to configure the mappings between the source tables and destination Kafka topics.

iv.

v. View the mapping progress, source tables, and mapped destination topics.



One-click real-time synchronization to Kafka

☑ Select Synchroniza... ☑ Configure Network ... ☑ Set Synchronizatio... 4 Set Destination Topic 5 Set Resources for S...

4.1 Basic Configurations of Destination Topic

* Destination: zn_kafka

□ Source tables without primary keys can be synchronized.

☑ Send heartbeat record

Refresh source table and Kafka Topic mapping

Execution progress:100% Success: 2 tables Failure: 0 tables Total: 2 tables

Search by source database name: Search by source database name

Search by source table: Search by source table

□	Serial Number	Source Database Name	Source Table	Synchronize primary key	Kafka Topic (A total of 2 data)	Topic creation method	Actions
□	1	chengdu_db	holostreamx	id (Database primary key)	chengdu_db_holostreamx	Create Topic	Edit additional fields Remove Table
□	2	chengdu_db	holostreamx1	id (Database primary key)	chengdu_[illegible]ostreamx1	Create Topic	Edit additional fields Remove Table

Batch Edit Additional Fields in Destination Topic

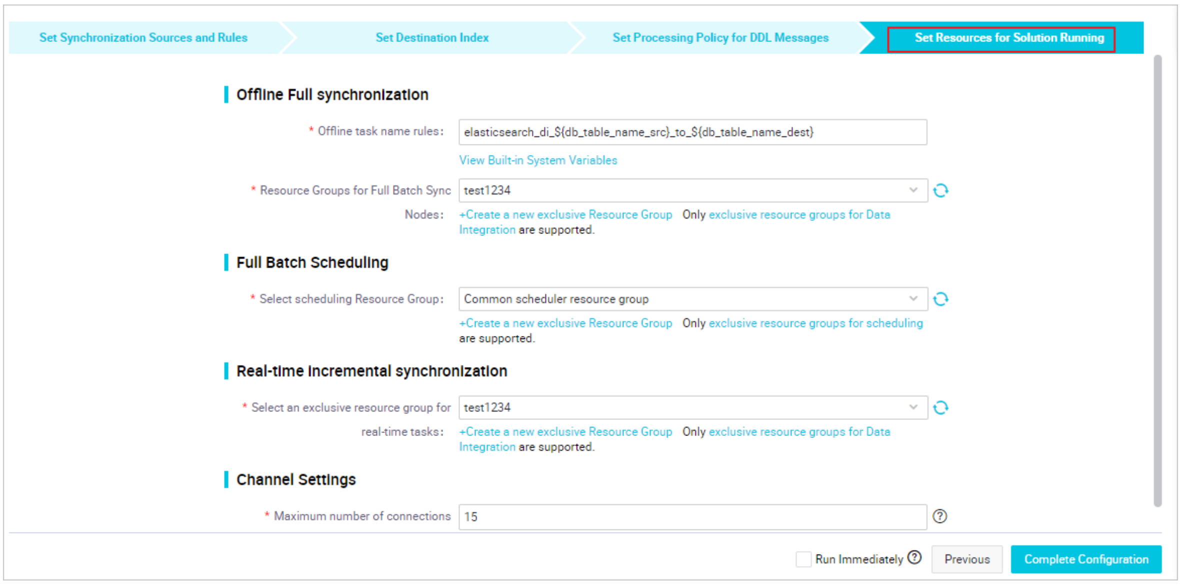
Batch Remove

Serial number	Description
1	The progress of mapping the source tables to destination tables. Note The mapping may require an extended period of time if you want to synchronize data from a large number of tables.

Serial number	Description
2	■ If you select **Source tables without primary keys can be synchronized.**, a source table that does not contain a primary key can be synchronized to the destination. However, duplicate data may exist if you perform data synchronization. ■ If you select **Send heartbeat record**, the real-time sync node writes a record that contains the current timestamp to Kafka every 5 seconds. This way, you can view the updates of the timestamp for the latest record written to Kafka and check the progress of the data synchronization even if no new records are written to Kafka.
3	■ If the tables in the source database contain primary keys, the system removes duplicate data based on the primary keys during the synchronization. ■ If you select **Source tables without primary keys can be synchronized.** and the source table does not contain a primary key, click the ✎ icon to specify a primary key. You can select one or more columns to serve as the primary key. The values of the one or more columns are used to remove duplicate data when you perform data synchronization.
4	The method that is used to create a destination topic. Valid values: **Use Existing Topic** and **Create Topic**.
5	The value in the Kafka Topic column varies with the value that you set for **Topic creation method**. ■ If you set the **Topic creation method** parameter to **Use Existing Topic**, you can select the destination topic from the drop-down list in the **Kafka Topic** column. ■ If you set the **Topic creation method** parameter to **Create Topic**, the name of the topic that is automatically created appears in the Kafka Topic column. You can click the automatically created topic to view and modify the name and description of the topic.
6	You can click **Batch Edit Additional Fields in Destination Topic** and add fields for multiple Kafka topics in the dialog box that appears. You can also click **Edit additional fields** in the **Actions** column to add additional fields for a single Kafka topic. **Note** The Batch Edit Additional Fields in Destination Topic feature takes effect only If you select **Create Topic** for the **Topic creation method** parameter.

4. Click **Next Step**.
5. Configure the resources required by the sync solution.

   In the **Set Resources for Solution Running** step, set the parameters as required.



Set Synchronization Sources and Rules | Set Destination Index | Set Processing Policy for DDL Messages | Set Resources for Solution Running

Offline Full synchronization

* Offline task name rules: elasticsearch_di_${db_table_name_src}_to_${db_table_name_dest}

View Built-in System Variables

* Resource Groups for Full Batch Sync Nodes: test1234

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

Full Batch Scheduling

* Select scheduling Resource Group: Common scheduler resource group

+Create a new exclusive Resource Group Only exclusive resource groups for scheduling are supported.

Real-time Incremental synchronization

* Select an exclusive resource group for real-time tasks: test1234

+Create a new exclusive Resource Group Only exclusive resource groups for Data Integration are supported.

Channel Settings

* Maximum number of connections 15

□ Run Immediately Previous Complete Configuration

- Offline Sync

Parameter	Description
Offline task name rules	The name of the batch sync node that is used to synchronize the full data of the source. After a sync solution is created, DataWorks first generates a batch sync node to synchronize full data, and then generates real-time sync nodes to synchronize incremental data.
Resource Groups for Full Batch Sync Nodes	The exclusive resource group for Data Integration that is used to run the batch sync node. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Scheduling Settings

Parameter	Description
Select scheduling Resource Group	The resource group for scheduling that is used to run the nodes. Only exclusive resource groups for Data Integration can be used to run sync solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Incremental Sync

Parameter	Description

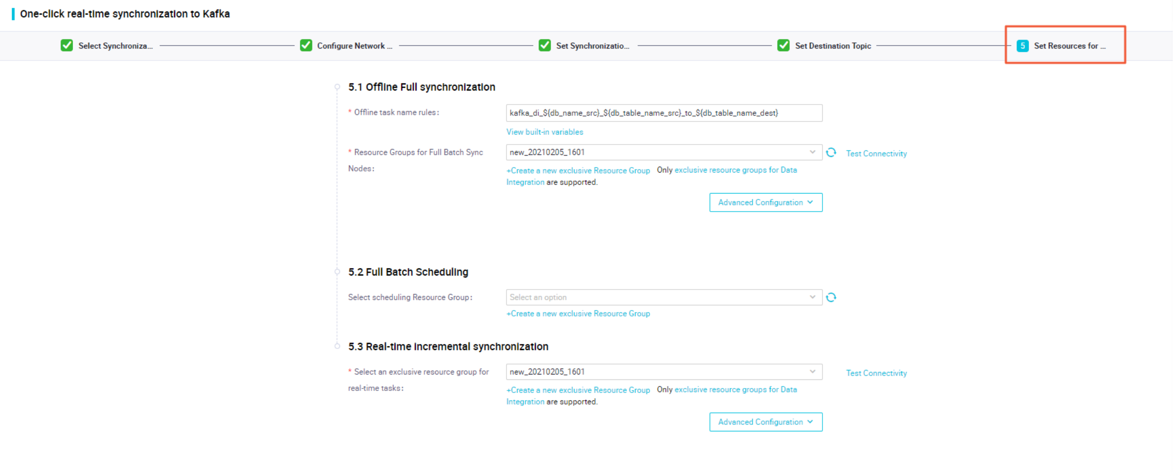
Parameter	Description
Resource Groups for Incremental Batch Sync Nodes	The exclusive resource group that is used to run the real-time sync nodes. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- **Channel Settings**

Parameter	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: **20**.

6. Configure the resources required by the sync solution.

   In the **Set Resources for Solution Running** step, set the parameters that are described in the following table.



One-click real-time synchronization to Kafka

5.1 Offline Full synchronization

* Offline task name rules: kafka_di_${db_name_src}_${db_table_name_src}_to_${db_table_name_dest}

* Resource Groups for Full Batch Sync Nodes: new_20210205_1601

Only exclusive resource groups for Data Integration are supported.

5.2 Full Batch Scheduling

Select scheduling Resource Group: Select an option

5.3 Real-time incremental synchronization

* Select an exclusive resource group for real-time tasks: new_20210205_1601

Only exclusive resource groups for Data Integration are supported.

- **Offline Sync**

Parameter	Description
Offline task name rules	The name of the batch sync node that is used to synchronize the full data of the source. After a sync solution is created, DataWorks first generates a batch sync node to synchronize full data, and then generates real-time sync nodes to synchronize incremental data.
Resource Groups for Full Batch Sync Nodes	The exclusive resource group for Data Integration that is used to run the batch sync node. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Scheduling Settings

Parameter	Description
Select scheduling Resource Group	The resource group for scheduling that is used to run the nodes. Only exclusive resource groups for Data Integration can be used to run sync solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. ? **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Incremental Sync

Parameter	Description
Resource Groups for Incremental Batch Sync Nodes	The exclusive resource group that is used to run the real-time sync nodes. Only exclusive resource groups for Data Integration can be used to run solutions. You can set this parameter to the name of the exclusive resource group for Data Integration that you purchased. For more information, see Plan and configure resources. ? **Note** If you do not have an exclusive resource group, click **Create a new exclusive Resource Group** to create one.

- Channel Settings

Parameter	Description
Maximum number of connections supported by source read	The maximum number of Java Database Connectivity (JDBC) connections that are allowed for the source. Specify an appropriate number based on the resources of the source. Default value: **20**.

7. Click **Complete Configuration** to return to the **Tasks** tab.
8. Find the sync solution from which you removed source tables and choose **More > Submit and Run** in the **Operation** column. In the **Submit and Run** message, click **OK** to run the solution.

   If you remove source tables from a sync solution that is running, the source tables are also removed from real-time sync nodes generated by the sync solution. After you submit and run the sync solution from which you removed source tables, the system continues to synchronize data at the time when the sync solution starts to be rerun.
9. View the removal details of the source tables.
   i.
   ii. In the **steps** section, find the **Display the increased/decreased table** node and click **Execution details** in the Status column.

       If the status of the **Display the increased/decreased table** node is **Succeeded**, the source tables are removed from the sync solution.
   iii. View the source tables that are removed from the sync solution.

4.9. View the status information about sync solutions

On the Overview of running status page, you can view the status information about specific sync solutions that is collected in a specified period of time. The information includes the status distribution of sync solutions, resource usage, status distribution of batch sync nodes and real-time sync nodes, synchronization rate, details of the synchronized data, and nodes with high latency. This helps you view the distribution and execution details of the nodes that are run and troubleshoot abnormal nodes. This way, the operations and maintenance (O&M) efficiency of sync nodes can be improved.

Go to the Overview of running status page

1. Log on to the DataWorks console.
2. In the left-side navigation pane, click **Workspaces**.
3. In the top navigation bar, select the region in which your workspace resides. Find the workspace and click **Data Integration** in the Actions column.
4. In the left-side navigation pane, choose **Sync Solutions > Overview of running status**.

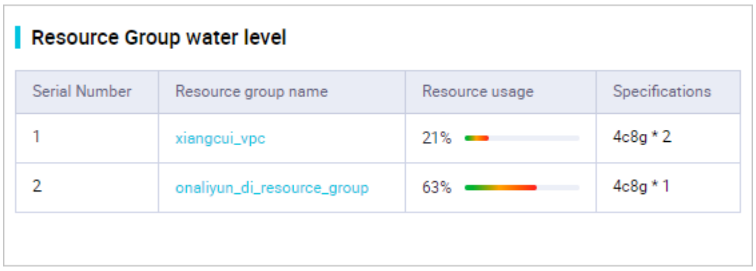
View the status information about sync solutions

You can view the status information about specific sync solutions that is collected in the specified period of time based on your business requirements.

- You can view the status information that is collected within the last week, last 48 hours, or last 24 hours. By default, status information that is collected within the last 24 hours is displayed.
- From the Task type drop-down list, you can select one or more of the following types: **One-click batch synchronization to Elasticsearch**, **One-click real-time synchronization to Elasticsearch**, **One-click real-time synchronization to MaxCompute**, and **One-click real-time synchronization to Hologres**. By default, all of the four types are selected.

On the Overview of running status page, you can view the status information in the following sections:

- The **Total operating state distribution** section displays the total number and status distribution of sync solutions, the number of successful sync solutions, and the number of failed sync solutions. The statistics are collected in the specified period of time. You can click a sector in the pie chart to go to the solution list page. On this page, you can view the successful or failed sync solutions and the execution details of the sync solutions. For more information about the execution details of a sync solution, see View the execution details of a sync solution.
- The **Resource Group water level** section displays the resource specifications and resource usage of the resource groups that are used by the current account. You can click the name of a resource group to go to the details page of the resource group. On the details page, you can view the basic information and resource usage of the resource group. For more information about resource groups, see View the resource usage of an exclusive resource group.



Resource Group water level

Serial Number	Resource group name	Resource usage	Specifications
1	xiangcui_vpc	21%	4c8g * 2
2	onaliyun_di_resource_group	63%	4c8g * 1

- The **Synchronize subtasks offline** section displays the number of batch sync nodes in specific sync solutions, synchronization rate, status distribution of the batch sync nodes, and details of the synchronized data. The statistics are collected in the specified period of time.

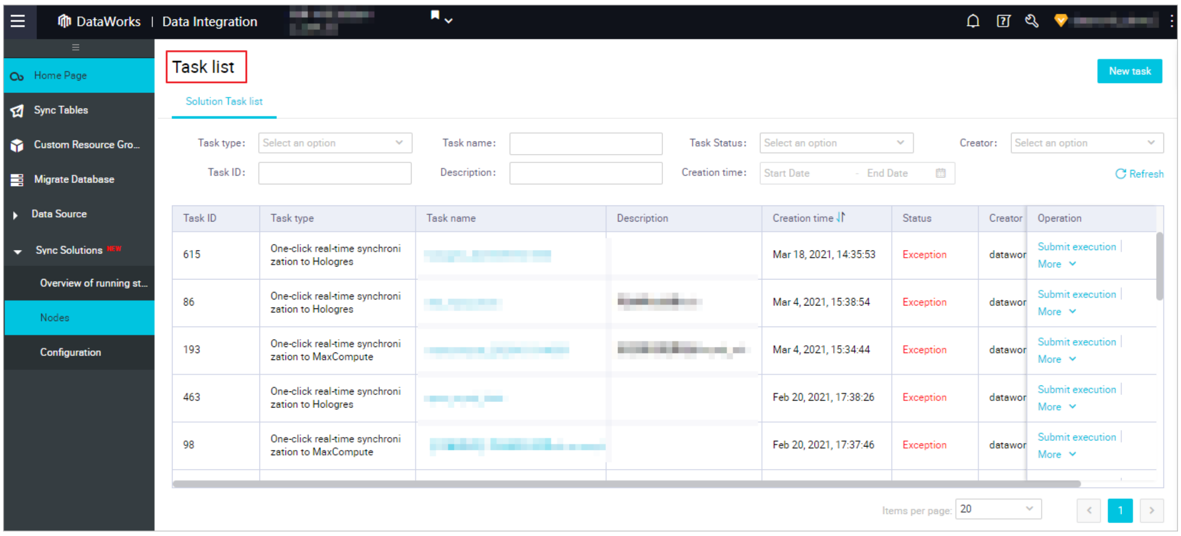
  > Note The statistics in the **Synchronize subtasks offline** section are updated on an hourly basis.

  - The statistics about status distribution are the distribution and numbers of the successful and failed nodes.
  - The Synchronize data subsection includes the following items:
    - Number of synchronization tasks: displays the number of batch sync nodes that are successful.
    - Amount of data synchronized: displays the amount of data that is synchronized by batch sync nodes that are successful or running.
    - Number of synchronization records: displays the number of data records that are synchronized by batch sync nodes.
- The **Synchronize subtasks in real time** section displays the number of real-time sync nodes in specific sync solutions, synchronization rate, status distribution of the real-time sync nodes, and top 10 nodes with the highest latency. You can click the name of a node to go to the O&M page and view the details of the node.

View the execution details of a sync solution

1. In the left-side navigation pane of the Data Integration page, choose **Sync Solutions > Nodes**.

   On the Task list page, you can view the ID, type, name, description, creation time, status, and creator of each sync solution. You can also perform operations on each sync solution. You can filter the sync solutions based on different conditions.



Task list

Task type: Select an option
Task name:
Task Status: Select an option
Creator: Select an option
Task ID:
Description:
Creation time: Start Date - End Date

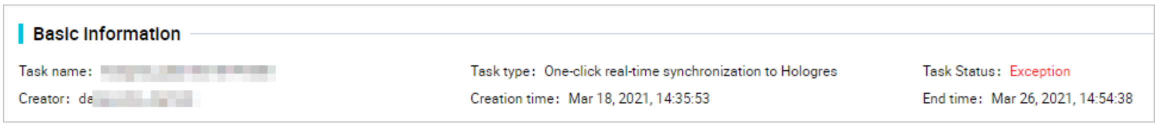
Task ID	Task type	Task name	Description	Creation time	Status	Creator	Operation
615	One-click real-time synchronization to Hologres	[redacted]		Mar 18, 2021, 14:35:53	Exception	datawor	Submit execution | More
86	One-click real-time synchronization to Hologres	[redacted]	[redacted]	Mar 4, 2021, 15:38:54	Exception	datawor	Submit execution | More
193	One-click real-time synchronization to MaxCompute	[redacted]	[redacted]	Mar 4, 2021, 15:34:44	Exception	datawor	Submit execution | More
463	One-click real-time synchronization to Hologres	[redacted]		Feb 20, 2021, 17:38:26	Exception	datawor	Submit execution | More
98	One-click real-time synchronization to MaxCompute	[redacted]		Feb 20, 2021, 17:37:46	Exception	datawor	Submit execution | More

Items per page: 20

2. On the Task list page, find the sync solution whose execution details you want to view. In the **Operation** column, click **More** and select **Execution details**.

   On the Execution details page, you can view the following information:

   - The **Basic Information** section displays the information about the sync solution, such as the status and execution time.

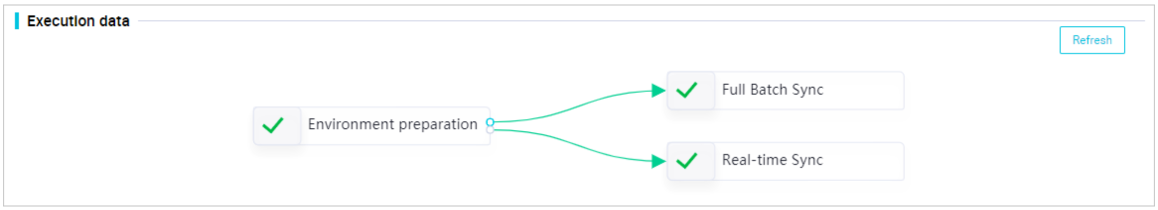
     Basic Information

     Task name: [redacted]
     Task type: One-click real-time synchronization to Hologres
     Task Status: Exception
     Creator: de[redacted]
     Creation time: Mar 18, 2021, 14:35:53
     End time: Mar 26, 2021, 14:54:38

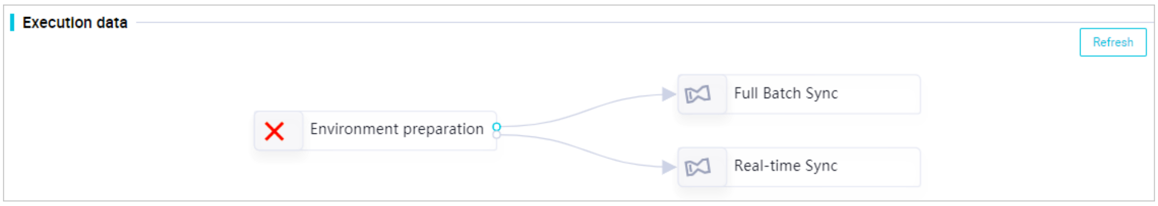
   - The **Execution data** section displays the statuses of the **Environment preparation**, **Full Batch Sync**, and **Real-time Sync** nodes. You can check whether the nodes are run as expected based on their status. This way, you can troubleshoot the issues of the sync solution at the earliest opportunity. A sync node can be in one of the following states:

     - If the ✓ icon is displayed, the node is successful.
     - If the ✗ icon is displayed, the node failed.
     - If the ⌛ icon is displayed, the node waits to be run.

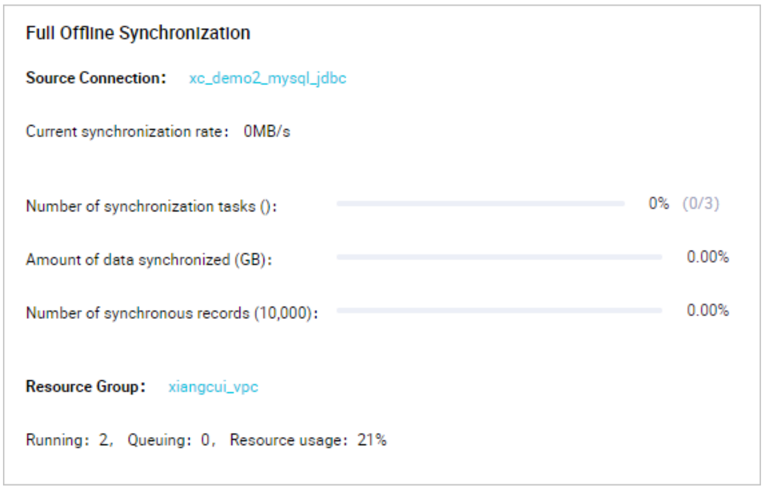
     Example 1: The following figure shows that the sync solution is successful.

     Example 2: The following figure shows that the **Environment preparation** node failed. In this case, the **Full Batch Sync** and **Real-time Sync** nodes are blocked and can only wait for execution and the sync solution failed. You can troubleshoot issues for the sync solution based on the status of its nodes at the earliest opportunity.

- The **Full Offline Synchronization** section displays the information about the **Full Batch Sync** node in the sync solution. The information includes the data source, current synchronization rate, details of the synchronized data, and details of the resource group that is used in batch synchronization.
  - The Synchronize data subsection includes the following items:
    - Number of synchronization tasks: displays the number of batch sync nodes that are successful.
    - Amount of data synchronized: displays the amount of data that is synchronized by batch sync nodes that are successful or running.
    - Number of synchronization records: displays the number of data records that are synchronized by batch sync nodes.
  - For more information about resource groups, see View the resource usage of an exclusive resource group.



Full Offline Synchronization

Source Connection: xc_demo2_mysql_jdbc

Current synchronization rate: 0MB/s

Number of synchronization tasks (): 0% (0/3)

Amount of data synchronized (GB): 0.00%

Number of synchronous records (10,000): 0.00%

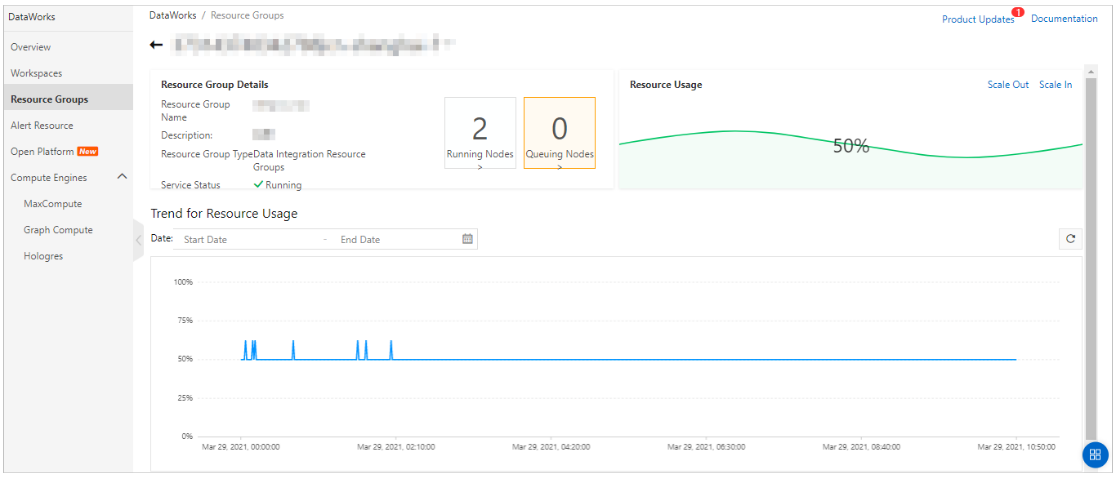
Resource Group: xiangcui_vpc

Running: 2, Queuing: 0, Resource usage: 21%

To view the details of the resource group in this section, perform the following steps:

a. Click the name of the resource group.

b. In the **Full offline synchronization details** dialog box, click **Details** to go to the details page of the resource group. On the details page, view the basic information, resource usage, and resource usage trend of the resource group.



Resource Group Details

Resource Group Name:

Description:

Resource Group Type: Data Integration Resource Groups

Service Status: ✓ Running

2 Running Nodes

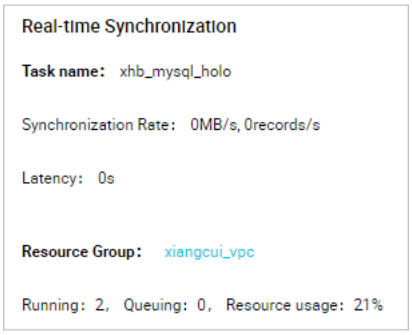
0 Queuing Nodes

Resource Usage

Trend for Resource Usage

Date: Start Date - End Date

- The **Real-time Synchronization** section displays the information about the **Real-time Sync** node in the sync solution. The information includes the name of the real-time sync node, current synchronization rate, synchronization latency, and resource group that is used in real-time synchronization.For more information about resource groups, see View the resource usage of an exclusive resource group.



Real-time Synchronization

Task name: xhb_mysql_holo

Synchronization Rate: 0MB/s, 0records/s

Latency: 0s

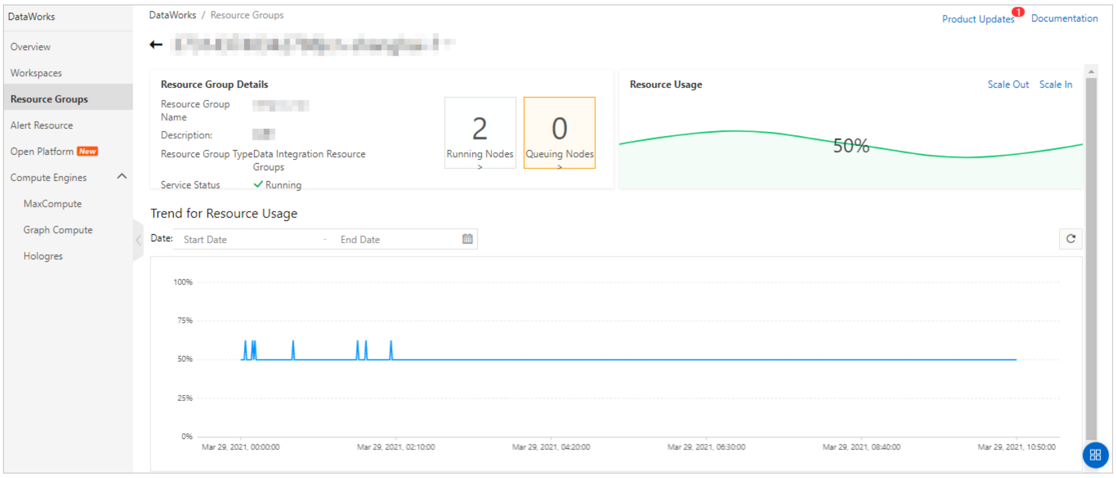
Resource Group: xiangcui_vpc

Running: 2, Queuing: 0, Resource usage: 21%

To view the details of the resource group in this section, perform the following steps:

a. Click the name of the resource group.

b. In the **Details of Real-time Synchronization** dialog box, click **Details** to go to the details page of the resource group. On the details page, view the basic information, resource usage, and resource usage trend of the resource group.



Resource Group Details

Resource Group Name: [redacted]

Description: [redacted]

Resource Group Type: Data Integration Resource Groups

Service Status: ✓ Running

2 Running Nodes

0 Queuing Nodes

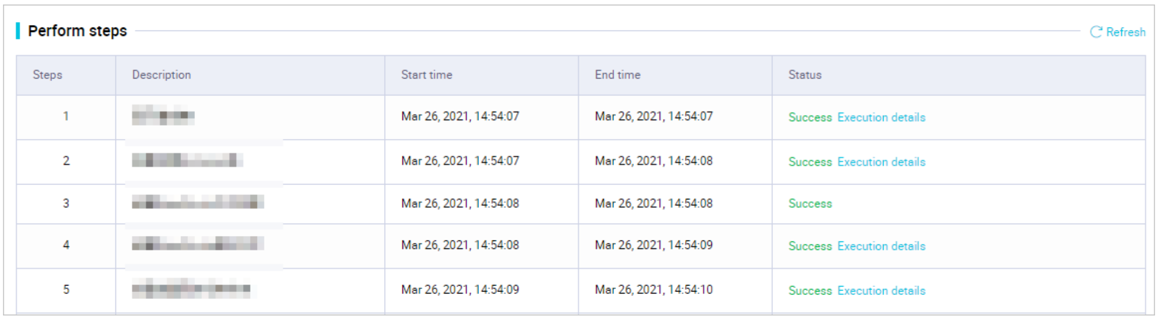
Resource Usage

50%

Trend for Resource Usage

Date: Start Date - End Date

- The **Perform steps** section displays all the steps in the sync solution from node creation to execution of batch sync nodes and real-time sync nodes. You can view the execution time and status of each step in this section.



Perform steps Refresh

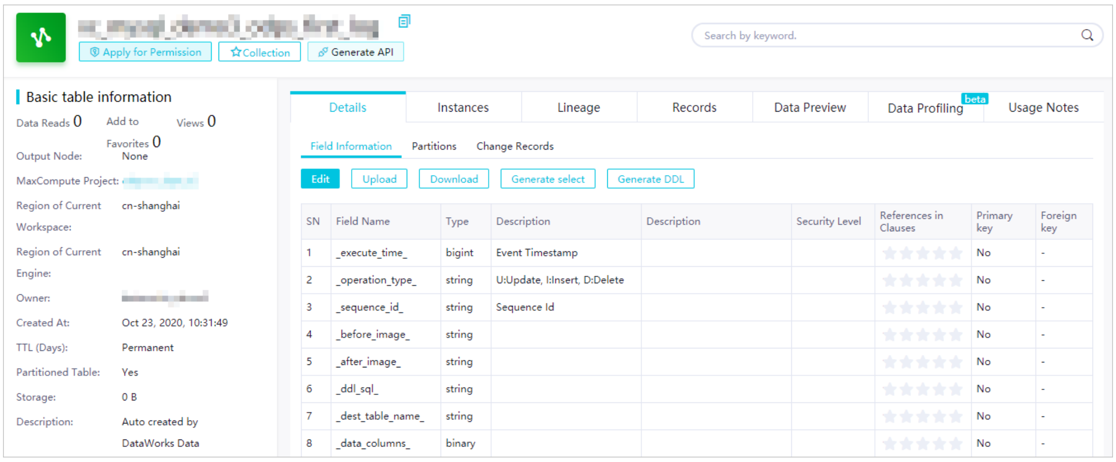
Steps	Description	Start time	End time	Status
1	[redacted]	Mar 26, 2021, 14:54:07	Mar 26, 2021, 14:54:07	Success Execution details
2	[redacted]	Mar 26, 2021, 14:54:07	Mar 26, 2021, 14:54:08	Success Execution details
3	[redacted]	Mar 26, 2021, 14:54:08	Mar 26, 2021, 14:54:08	Success
4	[redacted]	Mar 26, 2021, 14:54:08	Mar 26, 2021, 14:54:09	Success Execution details
5	[redacted]	Mar 26, 2021, 14:54:09	Mar 26, 2021, 14:54:10	Success Execution details

To view the execution details of s step, perform the following steps:

a. Find the step and click **Execution details** in the **Status** column.

b. In the dialog box that appears, click the name of a node to view the details of the node.

The following figure shows the details of the created log table in MaxCompute.



[redacted]

Basic table information

Data Reads 0 Add to Favorites 0 Views 0

Output Node: None

MaxCompute Project: [redacted]

Region of Current Workspace: cn-shanghai

Region of Current Engine: cn-shanghai

Owner: [redacted]

Created At: Oct 23, 2020, 10:31:49

TTL (Days): Permanent

Partitioned Table: Yes

Storage: 0 B

Description: Auto created by DataWorks Data

SN	Field Name	Type	Description	Description	Security Level	References in Clauses	Primary key	Foreign key
1	_execute_time_	bigint	Event Timestamp			☆☆☆☆☆	No	-
2	_operation_type_	string	U:Update, I:Insert, D:Delete			☆☆☆☆☆	No	-
3	_sequence_id_	string	Sequence Id			☆☆☆☆☆	No	-
4	_before_image_	string				☆☆☆☆☆	No	-
5	_after_image_	string				☆☆☆☆☆	No	-
6	_ddl_sql_	string				☆☆☆☆☆	No	-
7	_dest_table_name_	string				☆☆☆☆☆	No	-
8	_data_columns_	binary				☆☆☆☆☆	No	-

5.Appendixes

5.1. Connection configuration

5.1.1. Add an AnalyticDB for MySQL 2.0 data source

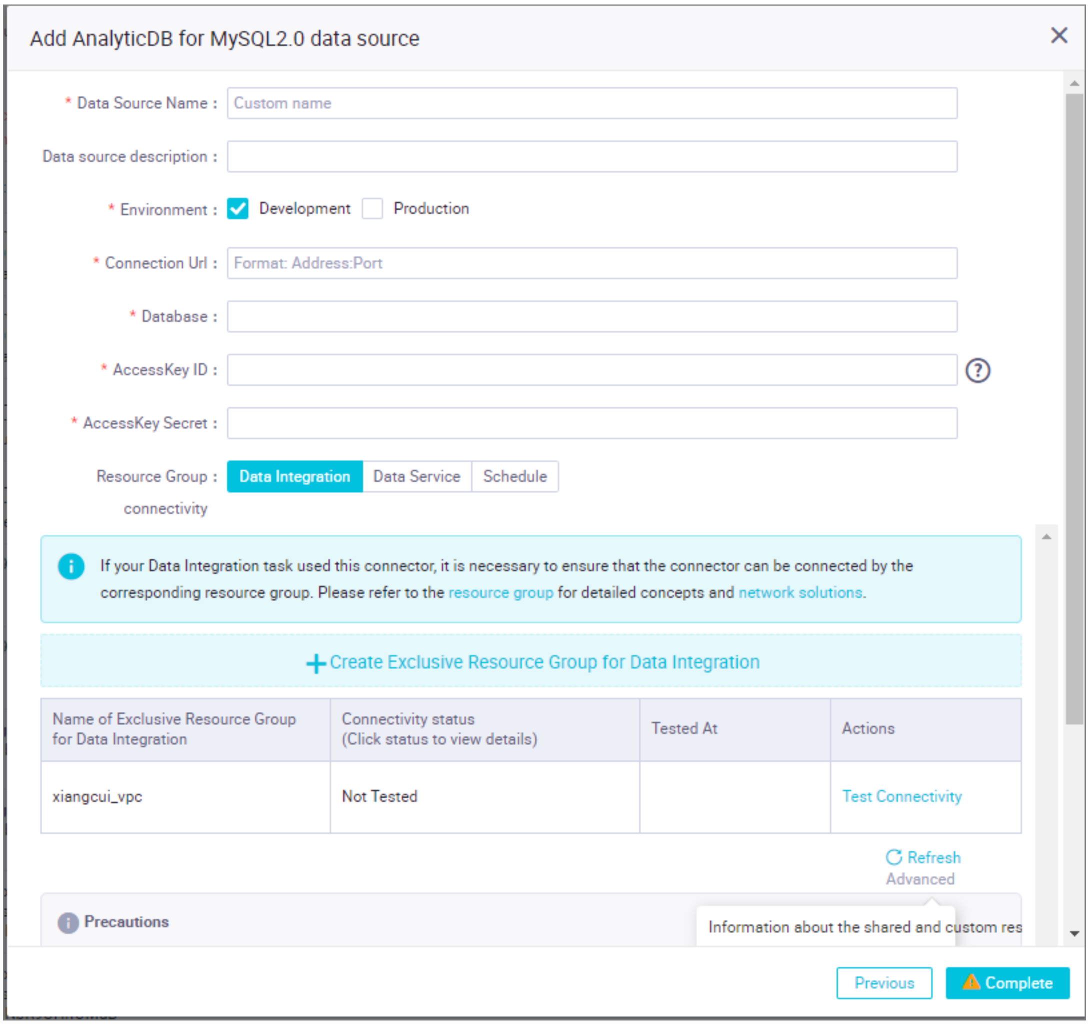
DataWorks provides AnalyticDB for MySQL 2.0 Reader and AnalyticDB for MySQL 2.0 Writer for you to read data from and write data to AnalyticDB for MySQL 2.0 data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for AnalyticDB for MySQL 2.0 data sources.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Procedure

1. Go to the **Data Source** page.
    i. Log on to the DataWorks console.
    ii. In the left-side navigation pane, click **Workspaces**.
    iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
    iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **AnalyticDB for MySQL 2.0** in the Relational Database section.
4. In the **Add AnalyticDB for MySQL 2.0 data source** dialog box, configure the parameters.



Add AnalyticDB for MySQL2.0 data source

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Connection Url : Format: Address:Port

* Database :

* AccessKey ID :

* AccessKey Secret :

Resource Group connectivity : Data Integration | Data Service | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
xiangcui_vpc	Not Tested		Test Connectivity

Refresh

Advanced

Precautions

Information about the shared and custom res

Previous Complete

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Connection Url	The connection string of the AnalyticDB for MySQL 2.0 database, in the format of `Endpoint of AnalyticDB for MySQL 2.0:Port number`.
Database	The name of the AnalyticDB for MySQL 2.0 database.
AccessKey ID	The AccessKey ID of the account that you use to connect to the AnalyticDB for MySQL 2.0 database. You can view the AccessKey ID on the Security Management page.
AceessKey Secret	The AccessKey secret of the account that you use to connect to the AnalyticDB for MySQL 2.0 database.

5. Set **Resource Group connectivity** to **Data Integration**.

6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > Note
   >
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add an AnalyticDB for MySQL 2.0 data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure AnalyticDB for MySQL 2.0 Reader and AnalyticDB for MySQL 2.0 Writer..

5.1.2. Add an SQL Server data source

DataWorks provides SQL Server Reader and SQL Server Writer for you to read data from and write data to SQL Server data sources. You can use the codeless user interface (UI) or code editor to configure sync nodes for SQL Server data sources.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Procedure

1. Go to the **Data Source** page.

   i. Log on to the DataWorks console.

   ii. In the left-side navigation pane, click **Workspaces**.

   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.

   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.

2. On the **Data Source** page, click **Add data source** in the upper-right corner.

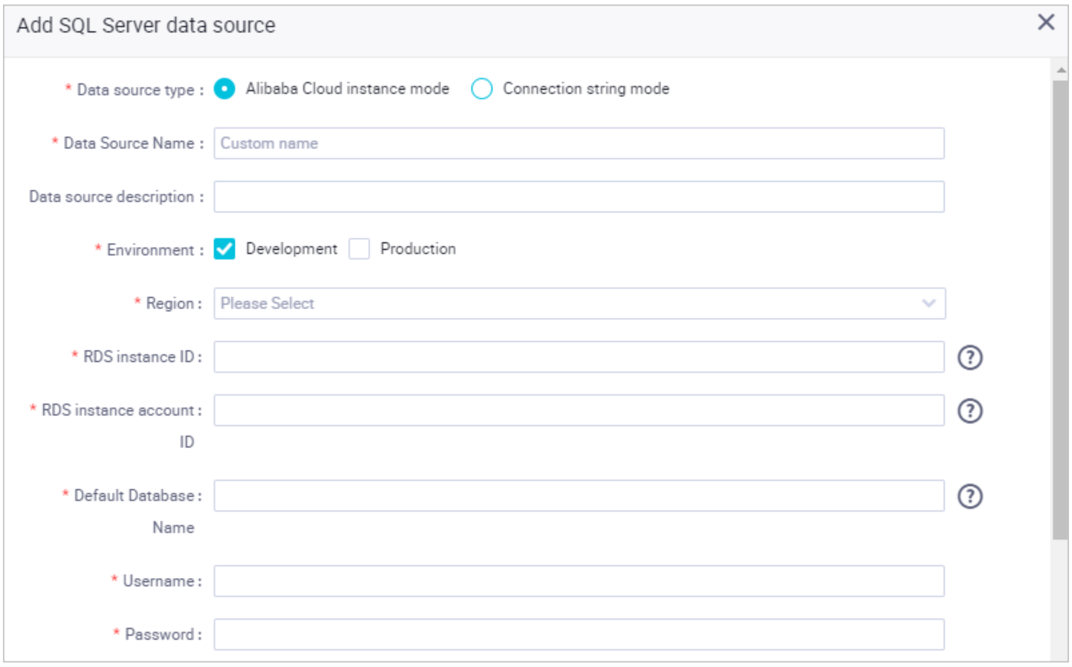
3. In the **Add data source** dialog box, click **SQLServer** in the Relational Database section.

4. In the **Add SQLServer data source** dialog box, set the parameters as required.

   i. Configure basic information for the SQL Server data source.

   You can use one of the following modes to add an SQL Server data source: **Alibaba Cloud instance mode** and **Connection string mode**.

   - The following table describes the parameters you can set if you add an SQL Server data source by using the **Alibaba Cloud instance mode**.



Add SQL Server data source

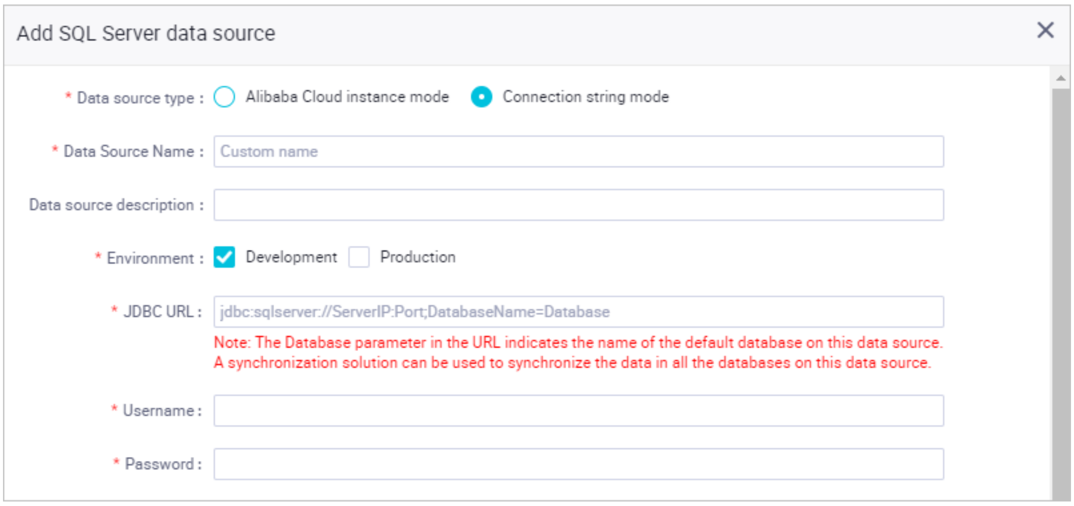
* Data source type : ● Alibaba Cloud instance mode ○ Connection string mode
* Data Source Name : Custom name
Data source description :
* Environment : ☑ Development □ Production
* Region : Please Select
* RDS instance ID :
* RDS instance account ID :
* Default Database Name :
* Username :
* Password :

Parameter	Description
Data source type	The mode in which the data source is added. Set this parameter to **Alibaba Cloud instance mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be up to 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Region	The region in which the data source resides.
RDS instance ID	The ID of the ApsaraDB RDS for SQL Server instance. You can log on to the ApsaraDB RDS console to view the ID of the instance.
RDS instance account ID	The ID of the Alibaba Cloud account that is used to purchase the ApsaraDB RDS for SQL Server instance. You can use the Alibaba Cloud account to log on to the DataWorks console, move the pointer over the profile picture in the upper-right corner, and then view the ID of the account.

Parameter	Description
Database name	The name of the default ApsaraDB RDS for MySQL database. The following descriptions provide instructions for you to configure a data synchronization node or solution that uses a MySQL data source: ■ When you configure a database-level real-time or batch data synchronization node or solution that uses a MySQL data source, you can select one or more databases on which you have access permissions in the ApsaraDB for MySQL instance. ■ If you select multiple databases when you configure a batch synchronization node, you must add a data source for each database.
User name	The username that you use to connect to the database.
Password	The password that you use to connect to the database. Do not use at signs *(@)* in your password.

Note You must add the IP addresses or CIDR blocks that you use to connect to the ApsaraDB RDS for SQL Server instance to a whitelist of the instance. For more information, see Configure a whitelist.

- The following table describes the parameters you can set if you add an SQL Server data source by using the **Connection string mode**.



Add SQL Server data source

* Data source type : ○ Alibaba Cloud instance mode ● Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* JDBC URL : jdbc:sqlserver://ServerIP:Port;DatabaseName=Database

Note: The Database parameter in the URL indicates the name of the default database on this data source.
A synchronization solution can be used to synchronize the data in all the databases on this data source.

* Username :

* Password :

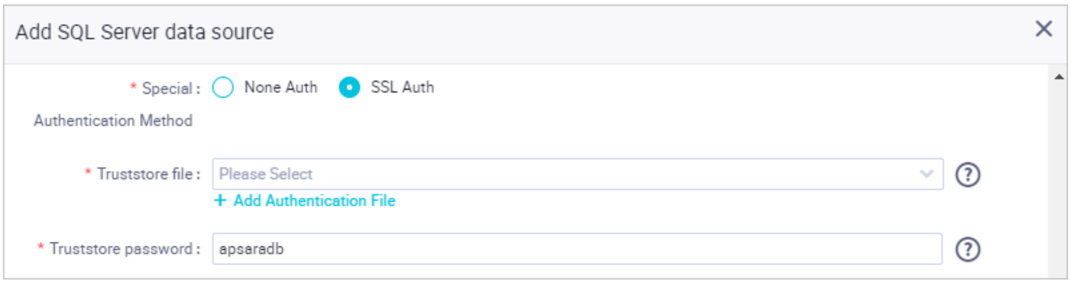
Parameter	Description
Data source type	The mode in which the data source is added. Set this parameter to **Connection string mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be up to 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database, in the format of `jdbc:sqlserver://ServerIP:Port;DatabaseName=Database`. You must replace Database with the name of the default database. However, when you configure a sync node, you can use all the databases in the ApsaraDB RDS instance.
User name	The username that you use to connect to the database.
Password	The password that you use to connect to the database.

ii. Configure authentication information for the SQL Server data source.

Third-party authentication mechanisms are used to perform strict identity authentication on users and services. These mechanisms prevent untrusted applications or services from accessing data and improve data security during data synchronization. DataWorks allows you to use a third-party authentication mechanism when you add a data source. If you enable identity authentication, only the trusted applications and services can access data sources. To enable SSL-based authentication, set the **Special Authentication Method** parameter to **SSL Auth**.

> **Note** If you enable SSL-based authentication, you must upload the required authentication files on the Authentication File Management page of the DataWorks console. For more information about how to upload and reference authentication files, see Upload and reference an authentication file. If you do not need to perform identity authentication on applications or services, you can set the **Special Authentication Method** parameter to **None Auth** when you add the SQL Server data source.

The following figure shows the parameters you must set if you enable SSL-based authentication.



Add SQL Server data source ×

* Special Authentication Method : ○ None Auth ● SSL Auth

* Truststore file : Please Select

+ Add Authentication File

* Truststore password : apsaradb

Parameter	Description
Truststore file	The truststore that is used to store specific trusted certificates. This certificate is authenticated when DataWorks accesses the SSL server. This ensures that only trusted applications and services can access data sources. You can lick **Add Authentication File** to upload a new truststore.
Truststore password	■ If your data source is an ApsaraDB RDS for MySQL, ApsaraDB RDS for SQL Server, or ApsaraDB RDS for PostgreSQL database, the password in the truststore is *apsaradb*. ■ If your data source is a self-managed RDS database, the value of the **Truststore password** parameter is a password that you specify.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > **Note**
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add an SQL Server data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure SQL Server Reader and SQL Server Writer. For more information, see SQLServer Reader and SQLServer Writer.

5.1.3. Add a MongoDB data source

MongoDB is a document-oriented database that is second only to Oracle and MySQL. DataWorks provides MongoDB Reader and MongoDB Writer for you to read data from and write data to MongoDB data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for MongoDB data sources.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

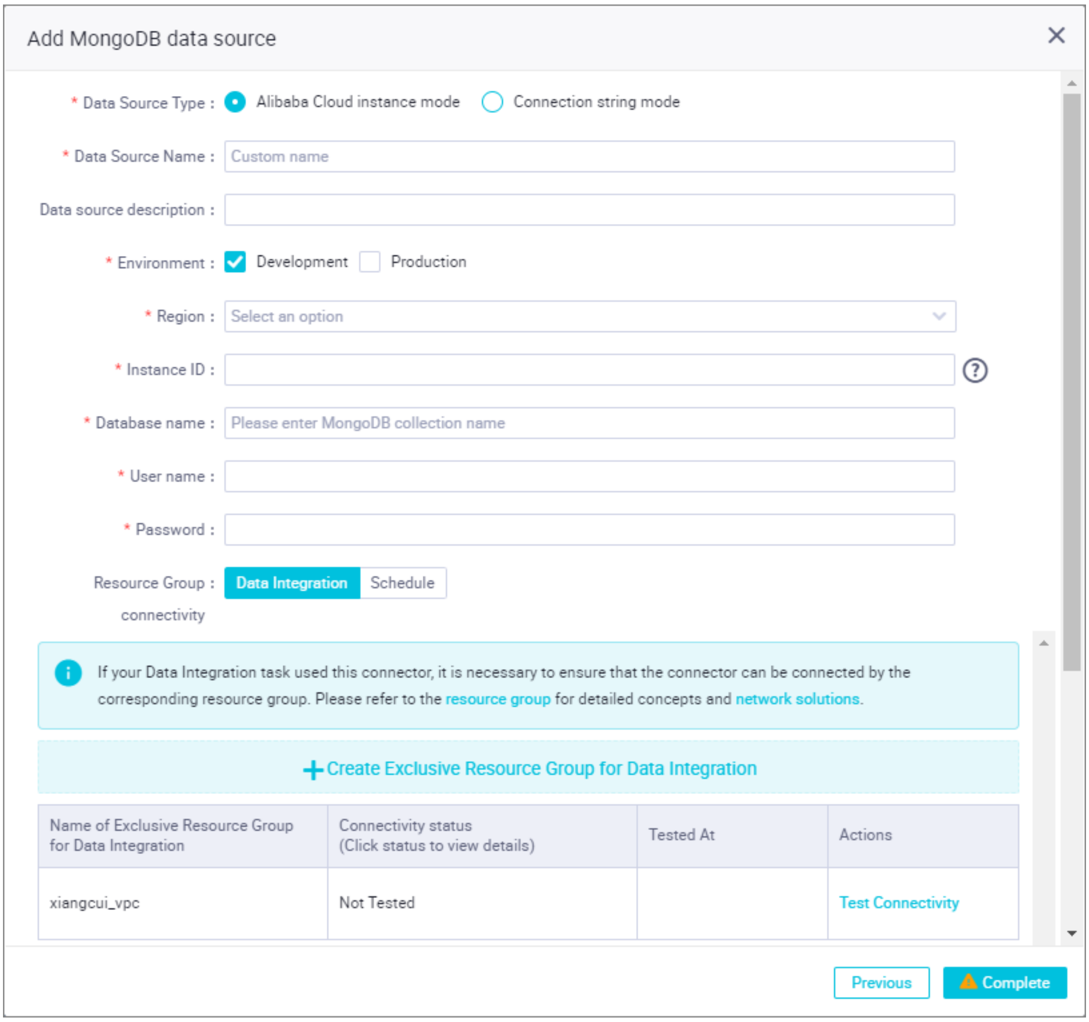
Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration**.
   iv. In the left-side navigation pane, choose **Data Source > Data Sources**.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **MongoDB** in the NoSQL section.
4. In the **Add MongoDB data source** dialog box, configure the parameters.

   You can use one of the following modes to add a MongoDB data source: **Alibaba Cloud Instance Mode** and **Connection String Mode**.

   > **Note** If the MongoDB data source that you want to add to a DataWorks workspace does not belong to the same Alibaba Cloud account as the workspace, you can add the MongoDB data source only in connection string mode.

   - **Alibaba Cloud Instance Mode**: In most cases, this mode is used to add a MongoDB data source that is deployed on the classic network. If the workspace and the MongoDB data source that you want to add reside in the same region, the MongoDB data source can be connected to the workspace over the classic network. If they reside in different regions, the connectivity between them over the classic network cannot be ensured.



Add MongoDB data source

- * Data Source Type : ● Alibaba Cloud instance mode ○ Connection string mode
- * Data Source Name : Custom name
- Data source description :
- * Environment : ☑ Development □ Production
- * Region : Select an option
- * Instance ID :
- * Database name : Please enter MongoDB collection name
- * User name :
- * Password :
- Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

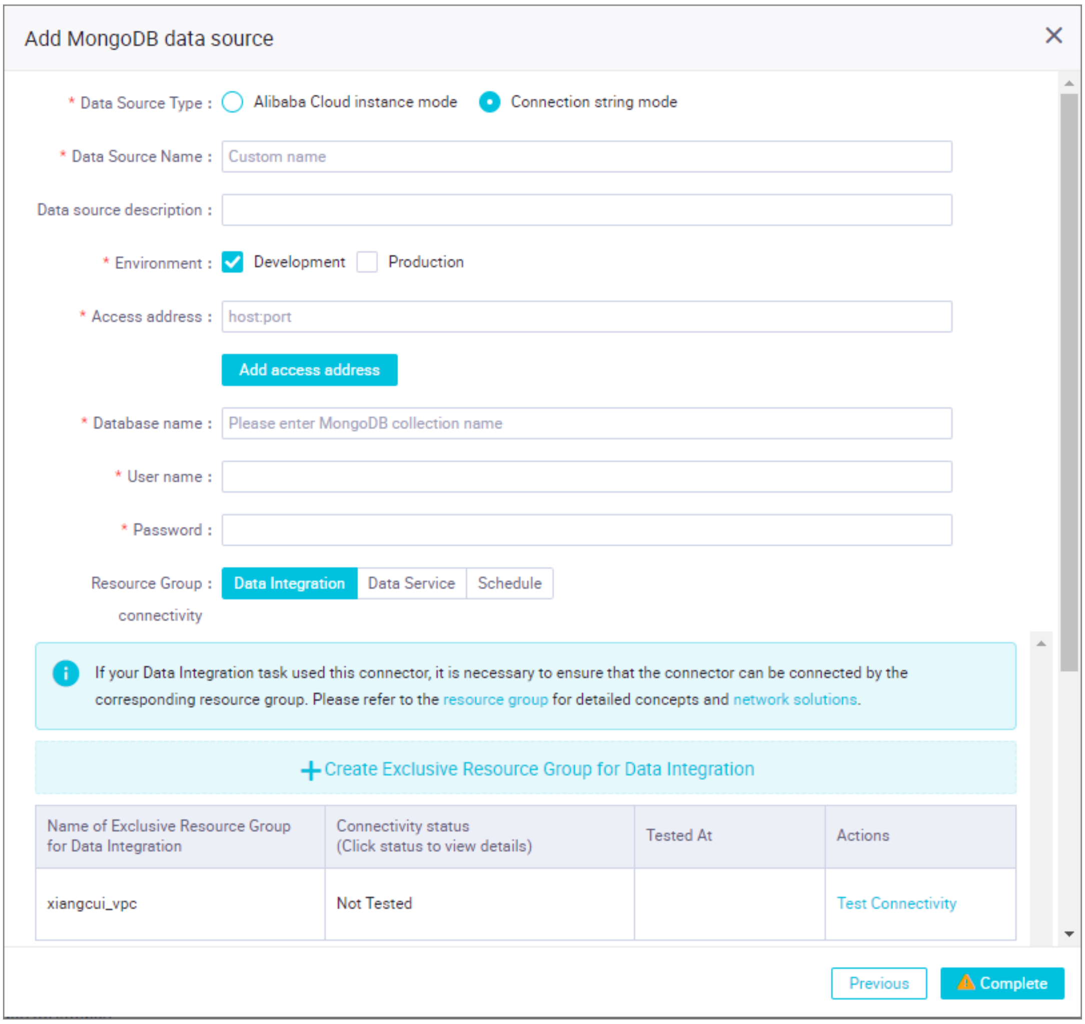
+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
xiangcui_vpc	Not Tested		Test Connectivity

Parameter	Description
Data Source Type	The mode in which the data source is added. Set this parameter to **Alibaba Cloud Instance Mode**. **Note** If you have not assigned the default role to Data Integration, log on to the Resource Access Management (RAM) console by using your Alibaba Cloud account and perform authorization. Then, refresh the configuration page.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data Source Description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Region	The region where the ApsaraDB for MongoDB instance resides.

Parameter	Description
Instance ID	The ID of the ApsaraDB for MongoDB instance. You can view the ID in the ApsaraDB for MongoDB console.
Database name	The name of the database that you created in the ApsaraDB for MongoDB console. You can create a database and specify a username and a password for the database in this console.
Username	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

- **Connection String Mode**: In most cases, this mode is used to add a MongoDB data source that is deployed on the Internet. Access to a MongoDB data source over the Internet may generate fees.


Add MongoDB data source

* Data Source Type : ○ Alibaba Cloud instance mode ● Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Access address : host:port

Add access address

* Database name : Please enter MongoDB collection name

* User name :

* Password :

Resource Group connectivity : Data Integration | Data Service | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
xiangcui_vpc	Not Tested		Test Connectivity

Previous | Complete

Parameter	Description
Data Source Type	The mode in which the data source is added. Set this parameter to **Connection String Mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data Source Description	The description of the data source. The description can be a maximum of 80 characters in length.

Parameter	Description
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. Note This parameter is displayed only when the workspace is in standard mode.
Address	The server address. Specify this parameter in the format of `Host address:Port number`. You can click **Add access address** to specify multiple addresses. Note If you specify multiple addresses, you must make sure that all the host IP addresses specified in the addresses are either public or private IP addresses.
Database name	The name of the database that you created in the ApsaraDB for MongoDB console.
Username	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

To add a MongoDB data source in connection string mode, perform the following steps:

a. Set Data Source Type to **Connection String Mode**.

b. In the **Add MongoDB data source** dialog box, configure the parameters. You must set the IP address of the host in the address specified by the Access address parameter to the private IP address of the data source.

c. Click **Complete** without testing the connectivity of the data source.

d. Create a custom resource group and use the resource group to run a synchronization node. For more information, see Create and use a custom resource group for Data Integration.

Notice

- ApsaraDB for MongoDB data sources can be connected only over the classic network.
- If an ApsaraDB for MongoDB instance is deployed in a virtual private cloud (VPC), you must set Data Source Type to Connection String Mode when you add the instance to DataWorks as a data source.
- If a MongoDB data source is deployed in a VPC, you cannot test the connectivity of the data source.

5. Set **Resource Group connectivity** to **Data Integration**.

6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   Note

   - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a MongoDB data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure MongoDB Reader and MongoDB Writer. For more information, see MongoDB Reader and MongoDB Writer.

5.1.4. Add a DataHub data source

DataHub offers a comprehensive data import scheme to support fast computing for large amounts of data.

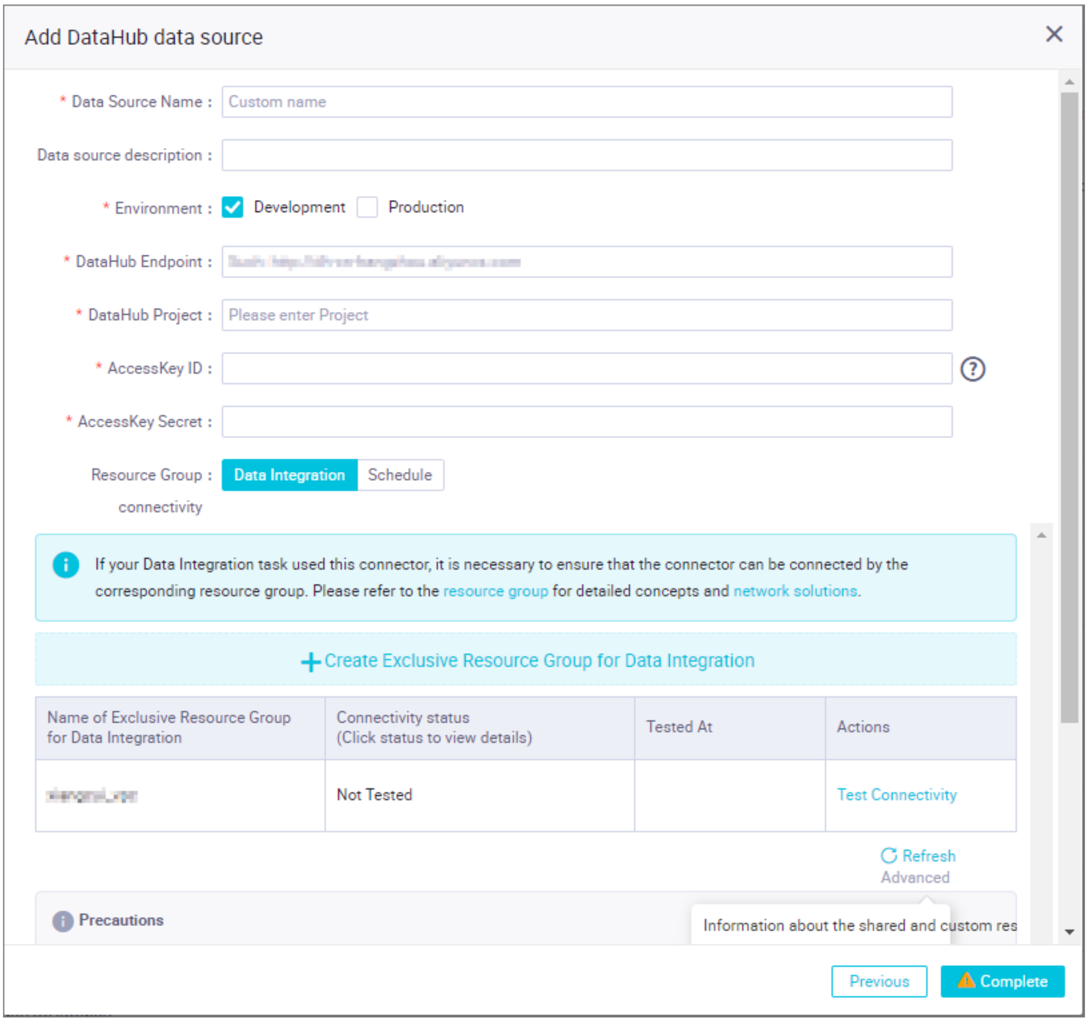
Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Data is synchronized based on the mappings between the data types of fields in DataHub and those in a specified service. DataHub supports only the following data types: **BIGINT**, **STRING**, **BOOLEAN**, **DOUBLE**, **TIMESTAMP**, and **DECIMAL**.

Procedure

1. Go to the **Data Source** page.
    i. Log on to the DataWorks console.
    ii. In the left-side navigation pane, click **Workspaces**.
    iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
    iv. In the left-side navigation pane of the Data Integration page, choose **Data Source** > **Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **DataHub** in the Big Data Storage section.
4. In the **Add DataHub data source** dialog box, set the parameters as required.



Add DataHub data source

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* DataHub Endpoint : [redacted]

* DataHub Project : Please enter Project

* AccessKey ID :

* AccessKey Secret :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+ Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
[redacted]	Not Tested		Test Connectivity

Refresh
Advanced

Precautions

Information about the shared and custom res

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
DataHub Endpoint	The endpoint of the DataHub project. The endpoint is automatically generated by DataHub based on system configurations.
DataHub Project	The ID of the DataHub project.
AccessKey ID	The AccessKey ID of the account that you use to connect to the DataHub project. You can view the AccessKey ID on the Security Management page.
AceessKey Secret	The AccessKey secret of the account that you use to connect to the DataHub project.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test**

connectivity in the Actions column.

A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

> Note
>
> - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
> - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a DataHub data source. You can proceed to the next tutorial. In the next tutorial, you will learn how to configure DataHub Writer. For more information, see DataHub Writer.

5.1.5. Add a DM data source

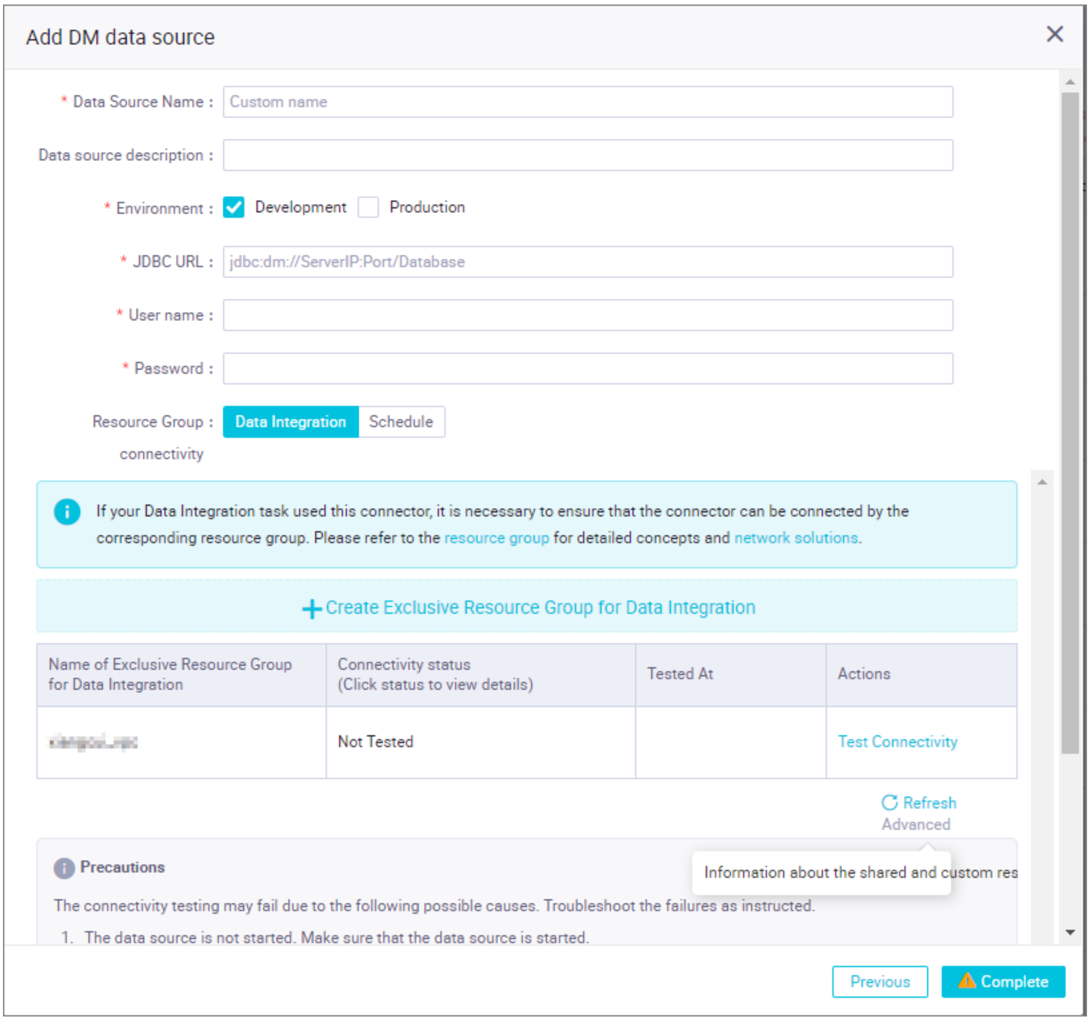
DataWorks provides DM Reader and DM Writer for you to read data from and write data to DM data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for DM data sources.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **DM** in the Relational Database section.
4. In the **Add DM data source** dialog box, configure the parameters.



Add DM data source

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* JDBC URL : jdbc:dm://ServerIP:Port/Database

* User name :

* Password :

Resource Group connectivity : Data Integration Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+ Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
[redacted]	Not Tested		Test Connectivity

Refresh
Advanced

Precautions

Information about the shared and custom res

The connectivity testing may fail due to the following possible causes. Troubleshoot the failures as instructed.

1. The data source is not started. Make sure that the data source is started.

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database, in the format of `jdbc:dm://ServerIP:Port/Database`.
User name	The username that you use to connect to the database.
Password	The password that you use to connect to the database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be

normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

> ⓘ Note
>
> - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
> - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

5.1.6. Add a DRDS data source

DataWorks provides DRDS Reader and DRDS Writer for you to read data from and write data to DRDS data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for DRDS data sources.

Context

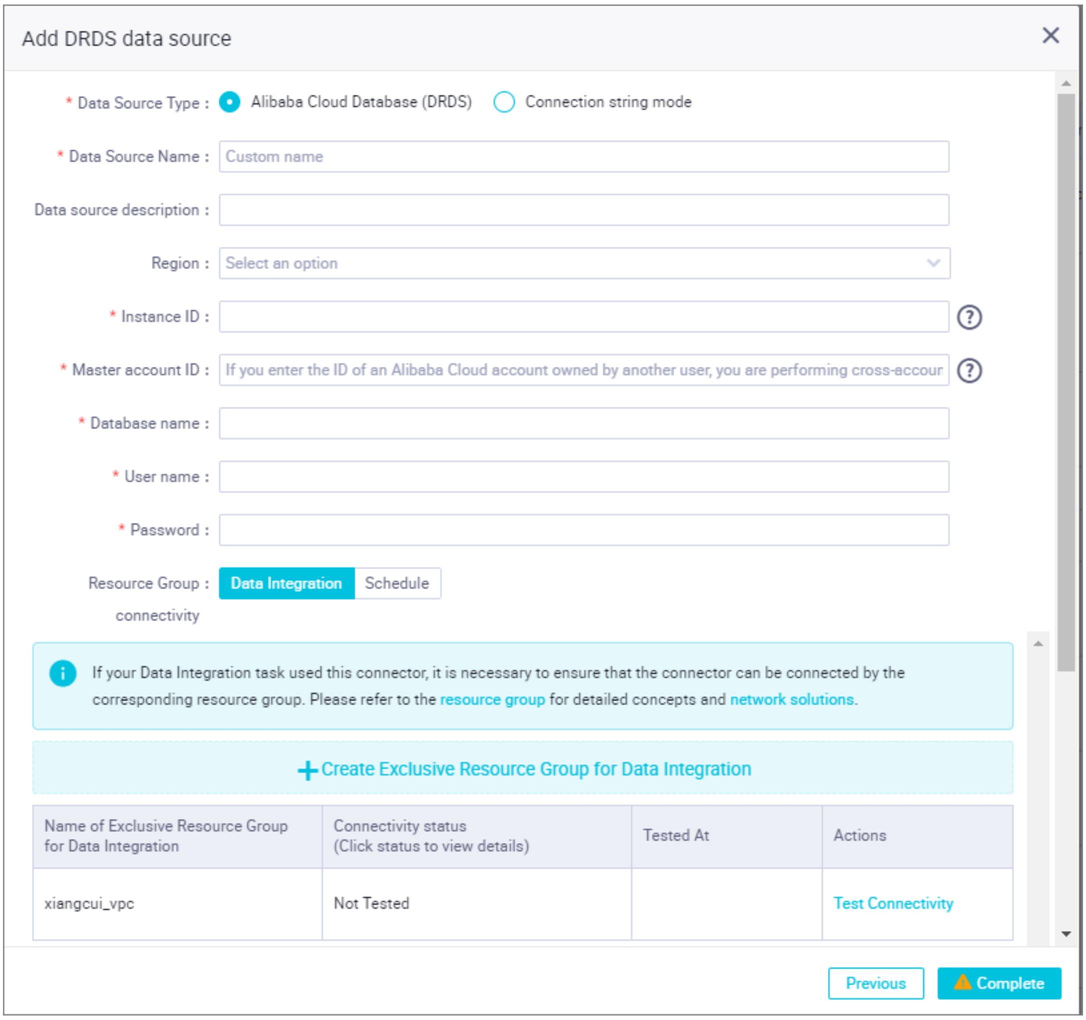
Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **DRDS** in the Relational Database section.
4. In the **Add DRDS data source** dialog box, configure the parameters.

   You can use one of the following modes to add a DRDS data source: **Alibaba Cloud Database (DRDS)** and **Connection string mode**.

   - The following table describes the parameters you must configure if you add a DRDS data source by using the **Alibaba Cloud database (DRDS) mode**.



Add DRDS data source

* Data Source Type : ● Alibaba Cloud Database (DRDS) ○ Connection string mode

* Data Source Name : Custom name

Data source description :

Region : Select an option

* Instance ID :

* Master account ID : If you enter the ID of an Alibaba Cloud account owned by another user, you are performing cross-accour

* Database name :

* User name :

* Password :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

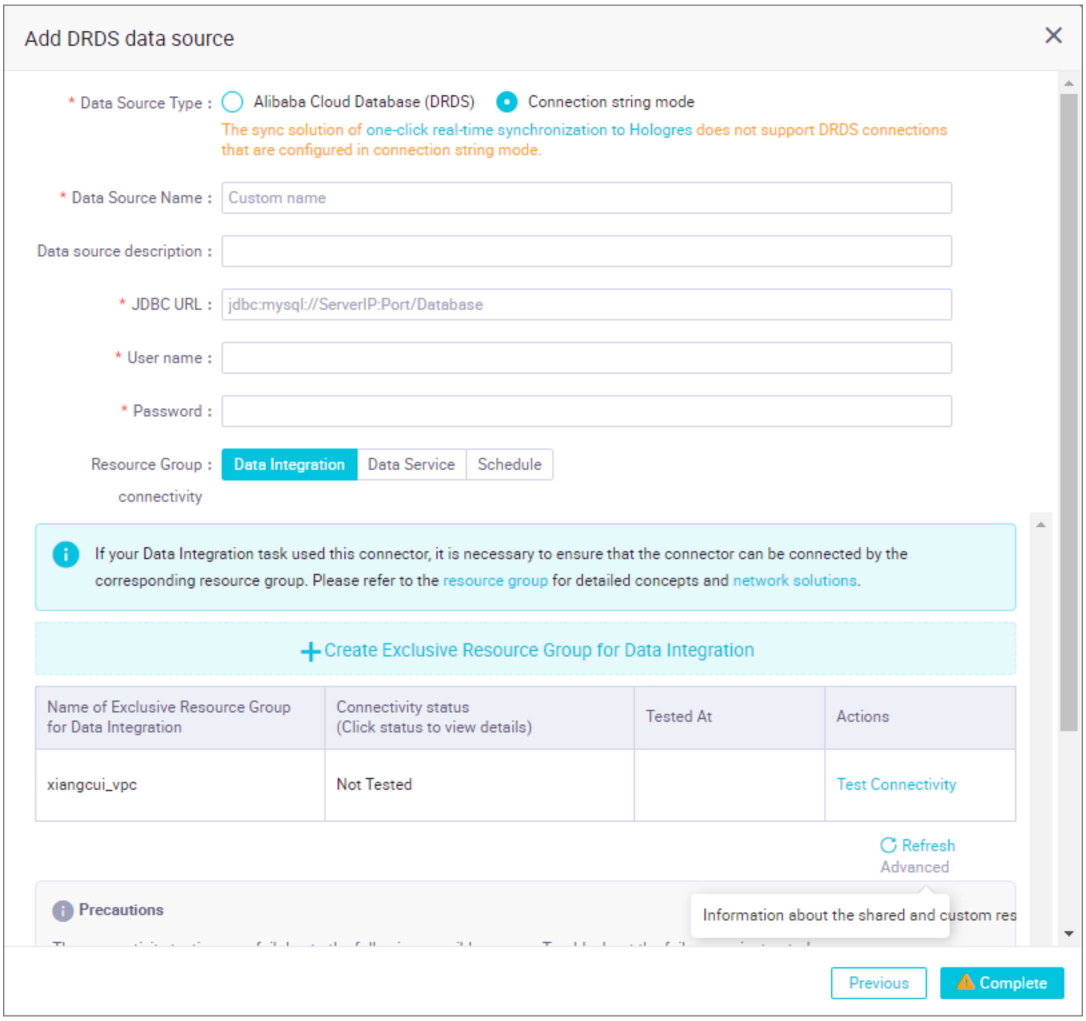
+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
xiangcui_vpc	Not Tested		Test Connectivity

Parameter	Description
Data source type	The type of the data source. Set this parameter to **Alibaba Cloud Database (DRDS)**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. Note This parameter is displayed only when the workspace is in standard mode.
Instance ID	The ID of the DRDS instance. You can log on to the DRDS console to view the ID.
Master account ID	The ID of the Alibaba Cloud account that is used to purchase the DRDS instance. To obtain the ID, perform the following steps: Log on to the DRDS console, move the pointer over the profile picture in the upper-right corner, and then select **Security Settings**. On the Security Settings page, you can view the value of the Account ID parameter, which is the ID of your Alibaba Cloud account.
Database name	The name of the database.

Parameter	Description
User name	The username that you use to connect to the database.
Password	The password that you use to connect to the database.

- The following table describes the parameters you must configure if you add a DRDS data source by using the **connection string mode**.



Add DRDS data source ✕

* Data Source Type : ○ Alibaba Cloud Database (DRDS) ● Connection string mode

The sync solution of one-click real-time synchronization to Hologres does not support DRDS connections that are configured in connection string mode.

* Data Source Name : Custom name

Data source description :

* JDBC URL : jdbc:mysql://ServerIP:Port/Database

* User name :

* Password :

Resource Group connectivity : Data Integration | Data Service | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
xiangcui_vpc	Not Tested		Test Connectivity

Parameter	Description
Data source type	The type of the data source. Set this parameter to **Connection string mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.

Parameter	Description
JDBC URL	The Java Database Connectivity (JDBC) URL of the database, in the format of `jdbc:mysql://ServerIP:Port/Database` .
User name	The username that you use to connect to the database.
Password	The password that you use to connect to the database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > Note
   >
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a DRDS data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure DRDS Reader and DRDS Writer. For more information, see DRDS Reader and DRDS Writer.

5.1.7. Add an FTP data source

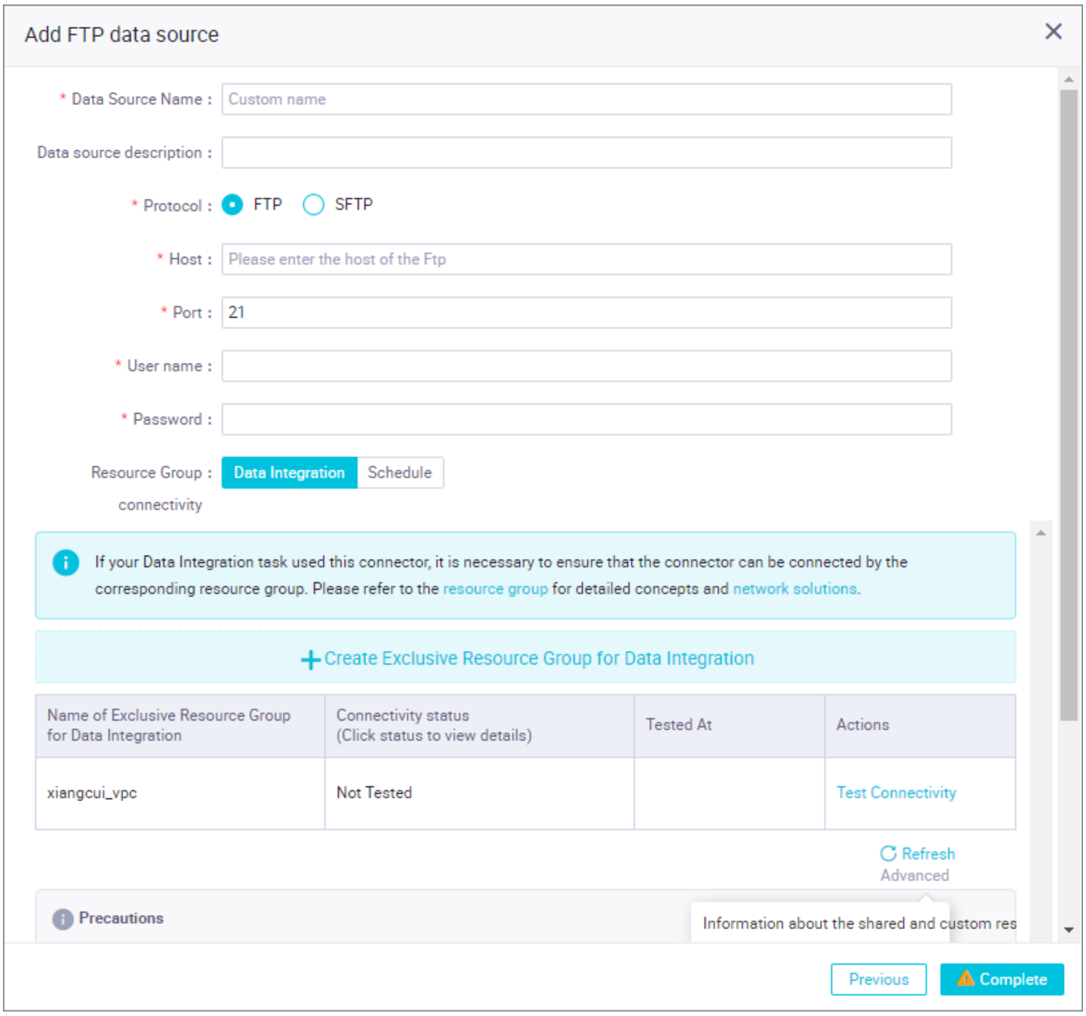
DataWorks provides FTP Reader and FTP Writer for you to read data from and write data to FTP data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for FTP data sources.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **FTP** in the Semi-structuredstorage section.
4. In the **Add FTP data source** dialog box, configure the parameters.



Add FTP data source

* Data Source Name : Custom name

Data source description :

* Protocol : ● FTP ○ SFTP

* Host : Please enter the host of the Ftp

* Port : 21

* User name :

* Password :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
xiangcui_vpc	Not Tested		Test Connectivity

Refresh

Advanced

Precautions

Information about the shared and custom res

Previous | Complete

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. Note This parameter is displayed only when the workspace is in standard mode.
Protocol	The protocol that is used by the FTP server. Valid values: FTP and SFTP.
Host	The IP address of the FTP server.
Port	The port number of the FTP server. The default port number for FTP is 21 and that for SFTP is 22.
User name	The username that you use to connect to the FTP server.
Password	The password that you use to connect to the FTP server.

5. Set **Resource Group connectivity** to **Data Integration**.

6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > Note
   >
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add an FTP data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure FTP Reader and FTP Writer. For more information, see FTP Reader and FTP Writer.

5.1.8. Add an HDFS data source

DataWorks provides HDFS Reader and HDFS Writer for you to read data from and write data to Hadoop Distributed File System (HDFS) data sources. This topic describes how to add an HDFS data source.

Context

Apsara File Storage for HDFS is not supported.

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information about the feature, see Isolate connections between the development and production environments.

If you use Object Storage Service (OSS) as the underlying storage, you must take note of the following items:

- The value of the defaultFS parameter must start with oss://. For example, the value can be `oss://IP:PORT` or `oss://nameservice`.
- You must configure the parameters that are required for connecting to OSS in the advanced parameters. The following code provides an example:

```
{
        "hadoopConfig":{
            "fs.oss.accessKeyId":"<yourAccessKeyId>",
                "fs.oss.accessKeySecret":"<yourAccessKeySecret>",
                "fs.oss.endpoint":"oss-cn-<yourRegion>-internal.aliyuncs.com"
        }
    }
```

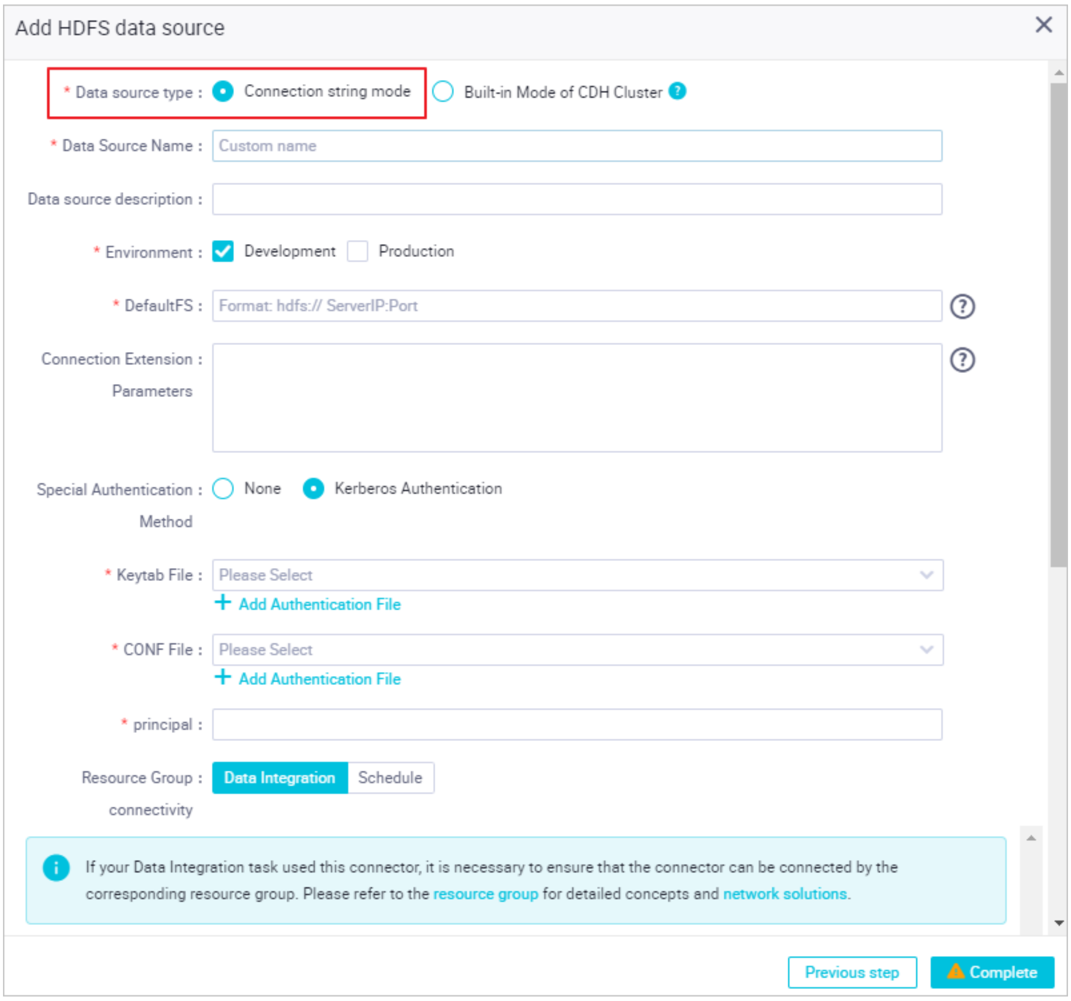
Procedure

1. Go to the **Data Source** page.
   1. Log on to the DataWorks console.
   2. In the left-side navigation pane, click **Workspaces**.
   3. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   4. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.

3. In the **Add data source** dialog box, click **HDFS** in the Semi-structuredstorage section.
4. In the **Add HDFS data source** dialog box, set the parameters as required.

   You can use one of the following modes to add an HDFS data source: **Connection String Mode** and **Built-in Mode of CDH Cluster**.

   - The following table describes the parameters that you can set if you add an **HDFS** data source by using the **Connection String Mode**.



Add HDFS data source

* Data source type : ● Connection string mode ○ Built-in Mode of CDH Cluster

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* DefaultFS : Format: hdfs:// ServerIP:Port

Connection Extension Parameters :

Special Authentication Method : ○ None ● Kerberos Authentication

* Keytab File : Please Select
+ Add Authentication File

* CONF File : Please Select
+ Add Authentication File

* principal :

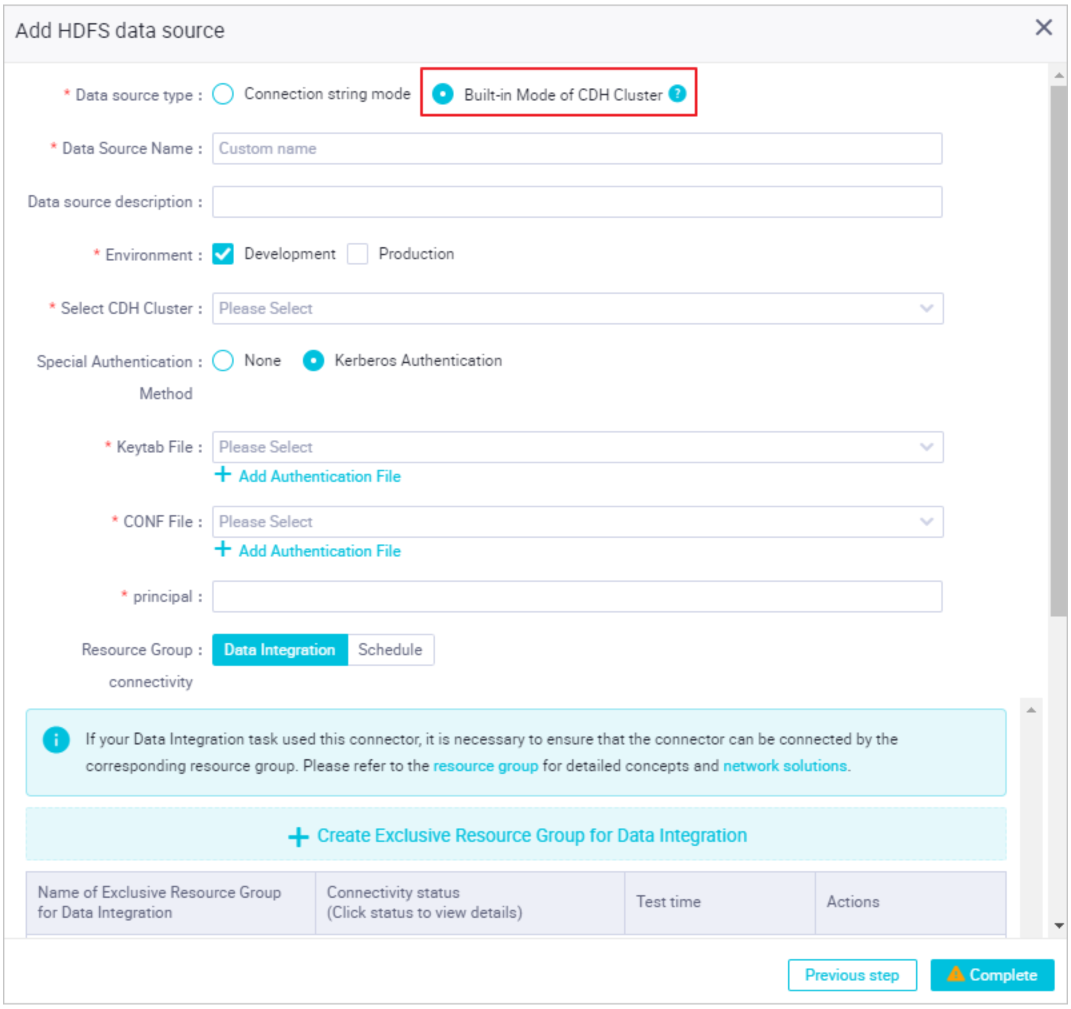
Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data Source Description	The description of the data source. The description can be up to 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. Note This parameter is displayed only when the workspace is in standard mode.
DefaultFS	The address of the NameNode in HDFS. Specify this parameter in the format of `hdfs://ServerIP:Port number`.

Parameter	Description
Connection Extension Parameters	The advanced parameters in hadoopConfig for HDFS Reader and HDFS Writer. You can configure the advanced parameters of Hadoop, such as those related to high availability.
Special Authentication Method	Specifies whether to enable identity authentication. Default value: **None**. You can alternatively set this parameter to **Kerberos Authentication**. For more information about Kerberos authentication, see Configure Kerberos authentication.
Keytab File	If you set the **Special Authentication Method** parameter to **Kerberos Authentication**, you must select the specified keytab from the Keytab File drop-down list. If no keytab is available, you can click **Add Authentication File** to upload a keytab.
CONF File	If you set the **Special Authentication Method** parameter to **Kerberos Authentication**, you must select the specified CONF file from the CONF File drop-down list. If no CONF file is available, you can click **Add Authentication File** to upload a CONF file.
principal	The Kerberos principal. Specify this parameter in the format of Principal name/Instance name@Domain name, such as ****/hadoopclient@**.***.

- The following table describes the parameters that you can set if you add an **HDFS** data source by using the **Built-in Mode of CDH Cluster**.



Add HDFS data source

* Data source type : ○ Connection string mode ● Built-in Mode of CDH Cluster

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Select CDH Cluster : Please Select

Special Authentication Method : ○ None ● Kerberos Authentication

* Keytab File : Please Select
+ Add Authentication File

* CONF File : Please Select
+ Add Authentication File

* principal :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+ Create Exclusive Resource Group for Data Integration
Name of Exclusive Resource Group for Data Integration
Connectivity status (Click status to view details)
Test time
Actions
Previous step
Complete

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data Source Description	The description of the data source. The description can be up to 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Select CDH Cluster	Select the specified CDH cluster from the drop-down list.
Special Authentication Method	Specifies whether to enable identity authentication. Default value: **None**. You can alternatively set this parameter to **Kerberos Authentication**. For more information about Kerberos authentication, see Configure Kerberos authentication.

Parameter	Description
Keytab File	If you set the **Special Authentication Method** parameter to **Kerberos Authentication**, you must select the specified keytab from the Keytab File drop-down list. If no keytab is available, you can click **Add Authentication File** to upload a keytab.
CONF File	If you set the **Special Authentication Method** parameter to **Kerberos Authentication**, you must select the specified CONF file from the CONF File drop-down list. If no CONF file is available, you can click **Add Authentication File** to upload a CONF file.
principal	The Kerberos principal. Specify this parameter in the format of Principal name/Instance name@Domain name, such as ****/hadoopclient@**.***.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   Note

   - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.
7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add an HDFS data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure HDFS Reader and HDFS Writer. For more information, see HDFS Reader and HDFS Writer.

5.1.9. Add a LogHub (SLS) data source

DataWorks provides LogHub (SLS) Reader and LogHub (SLS) Writer for you to read data from and write data to LogHub (SLS) data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for LogHub (SLS) data sources.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

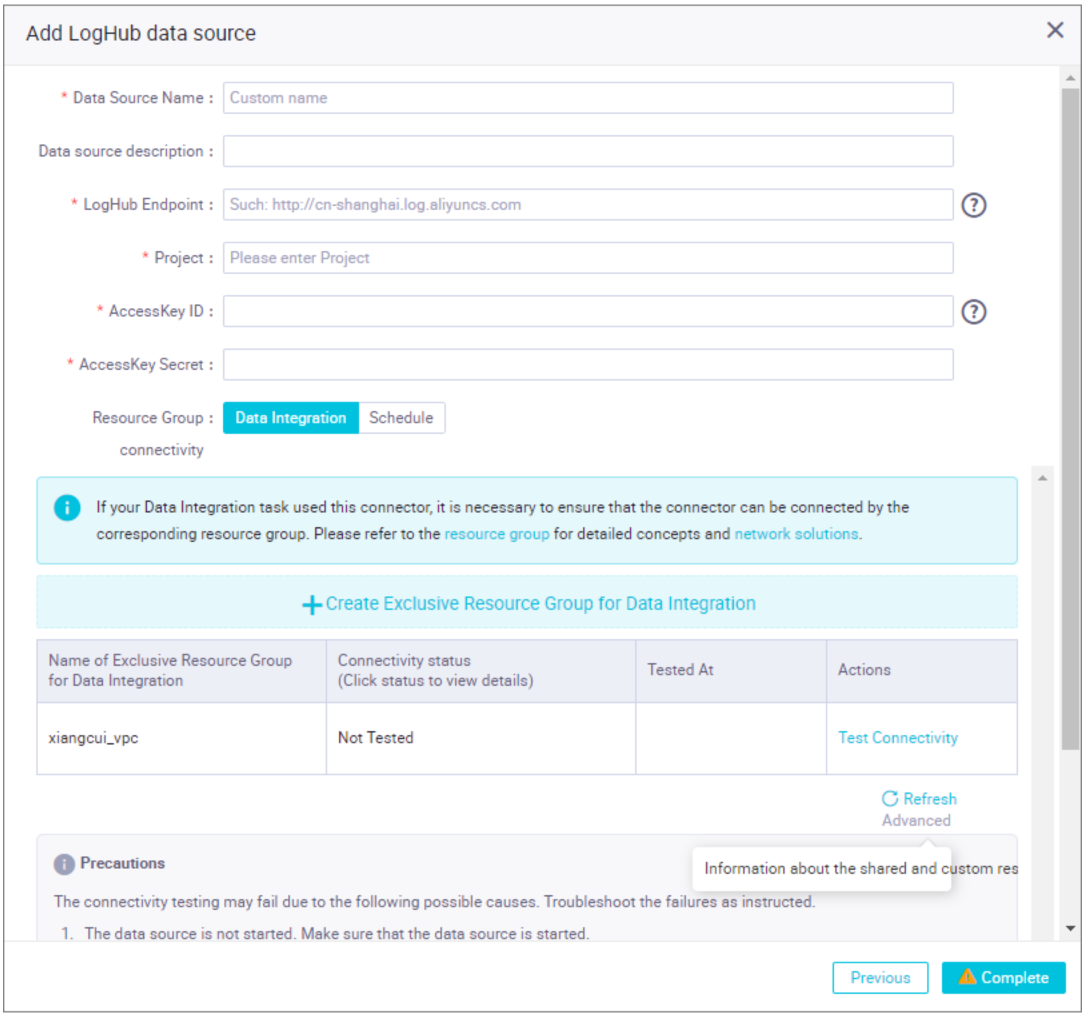
Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.

iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.

iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.

2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **LogHub** in the Message Queue section.
4. In the **Add LogHub data source** dialog box, configure the parameters.



Add LogHub data source

* Data Source Name : Custom name

Data source description :

* LogHub Endpoint : Such: http://cn-shanghai.log.aliyuncs.com

* Project : Please enter Project

* AccessKey ID :

* AccessKey Secret :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
xiangcui_vpc	Not Tested		Test Connectivity

Refresh

Advanced

Information about the shared and custom res

Precautions

The connectivity testing may fail due to the following possible causes. Troubleshoot the failures as instructed.

1. The data source is not started. Make sure that the data source is started.

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.

Parameter	Description
LogHub Endpoint	The endpoint of LogHub (SLS). Specify the endpoint in a format similar to `http://cn-shanghai.log.aliyun.com`. For more information, see Endpoints. Note Do not enter extra characters such as spaces or forward slashes (/) in the LogHub Endpoint field.
Project	The name of the LogHub (SLS) project.
AccessKey ID	The AccessKey ID of the Alibaba Cloud account that is used to log on to the Alibaba Cloud Management Console. You can go to the Security Management page to obtain the AccessKey ID.
AceessKey Secret	The AccessKey secret of the Alibaba Cloud account that is used to log on to the Alibaba Cloud Management Console. The secret is equivalent to the password that is used to log on to the DataWorks console. You can go to the Security Management page to obtain the AccessKey secret.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   Note

   - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.
7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a LogHub (SLS) data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure LogHub (SLS) Reader and LogHub (SLS) Writer. For more information, see LogHub (SLS) Reader and LogHub (SLS) Writer.

5.1.10. Add a Memcache data source

DataWorks provides Memcache Writer for you to write data to Memcache data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for Memcache data sources.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

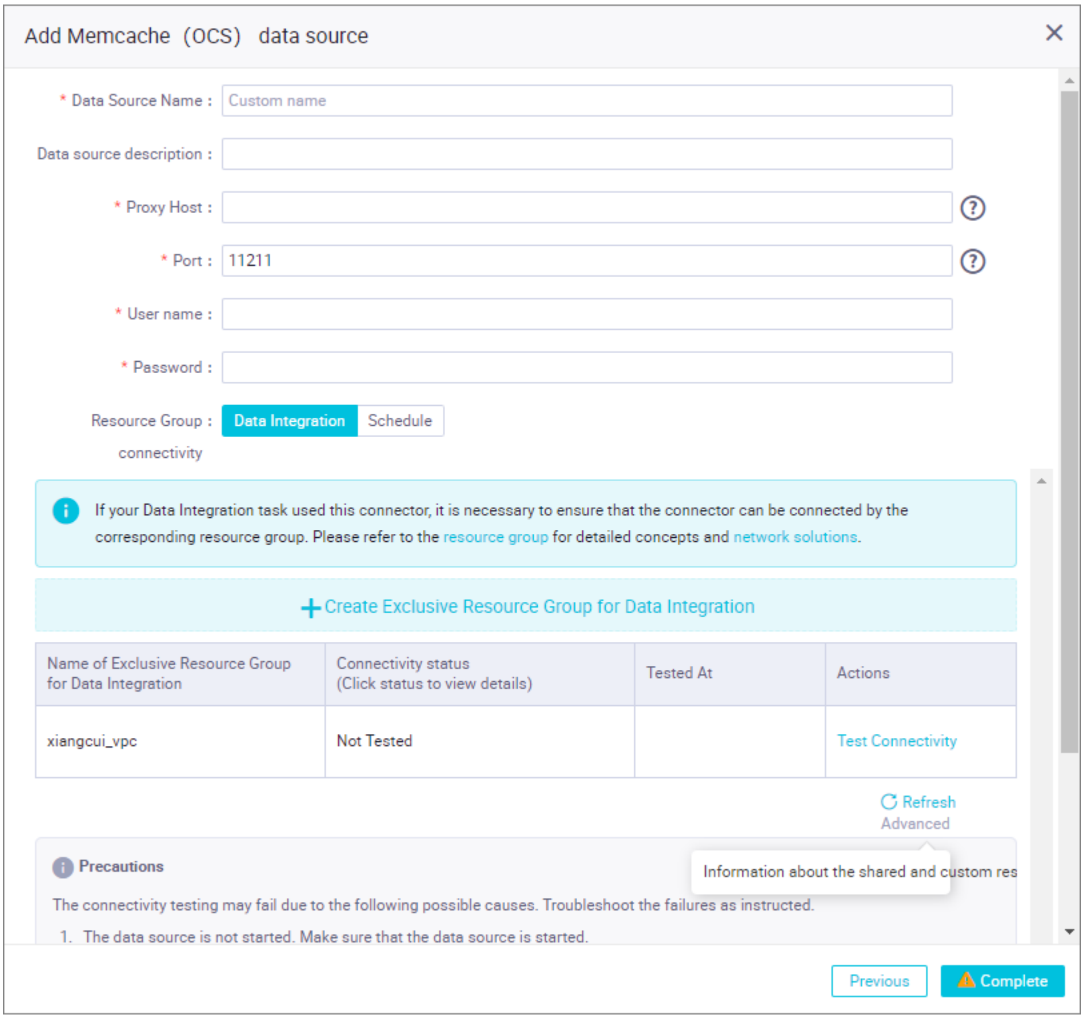
Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.

iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.

iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.

2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **Memcache (OCS)** in the NoSQL section.
4. In the **Add Memcache (OCS) data source** dialog box, configure the parameters.



Add Memcache（OCS）data source

* Data Source Name : Custom name

Data source description :

* Proxy Host :

* Port : 11211

* User name :

* Password :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
xiangcui_vpc	Not Tested		Test Connectivity

Refresh
Advanced

Information about the shared and custom res

Precautions

The connectivity testing may fail due to the following possible causes. Troubleshoot the failures as instructed.

1. The data source is not started. Make sure that the data source is started.

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Proxy Host	The IP address of the host or proxy host. You can view the IP address on the basic information page of the ApsaraDB for Memcache instance in the ApsaraDB for Memcache console.

Parameter	Description
Port	The port number that is used to connect to the ApsaraDB for Memcache instance. Default value: 11211.
User name	The username that you use to connect to the data source.
Password	The password that you use to connect to the data source.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > Note
   >
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a Memcache data source. You can proceed to the next tutorial. In the next tutorial, you will learn how to configure Memcache Writer. For more information, see Memcache Writer.

5.1.11. Add a MySQL data source

DataWorks provides MySQL Reader and MySQL Writer for you to read data from and write data to MySQL data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for MySQL data sources. This topic describes how to add a MySQL data source. This topic also describes the network environment and permissions that you must prepare before you add a data source.

Prerequisites

Before you add a data source, make sure that the following operations are performed:

- Prepare a data source: A MySQL data source is created.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you add the data source to DataWorks, connect the data source to an exclusive resource group for Data Integration based on your business requirements. After you connect the data source to the exclusive resource group for Data Integration, configure network access settings such as a vSwitch and a whitelist.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

  For more information about how to configure a whitelist, see Configure whitelists for data sources.
- Execute the following statement to check the version of your MySQL database. You can use a `MySQL 5.X` or `MySQL 8.X` database.

```
select version();
```

ⓘ **Note** Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.x` or `V8.x`. PolarDB for MySQL is not supported. If the MySQL version of your ApsaraDB RDS for MySQL database is not `V5.x` or `V8.x`, use an ApsaraDB RDS for MySQL database whose MySQL version is `V5.x` or `V8.x`. Otherwise, the data synchronization node fails to run.

- Prepare an account and grant permissions to the account.

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

  For more information, see Create and authorize an account.
- Enable the MySQL binary logging feature. This feature is required only for real-time data synchronization. For more information about real-time data synchronization, see Overview.

  If the source data source is a MySQL database, you must enable the binary logging feature. Binary logs record changes to all table schemas and modifications on table data. You can execute statements such as CREATE and ALTER to perform operations on table schemas. You can execute statements such as INSERT, UPDATE, and DELETE to perform operations on table data. You can use binary logs to view the change history of the database, back up incremental data and restore data in the database, and replicate data from the primary database to secondary databases.

  Formats of binary logs:

  - Statement: SQL statement-based replication. Binary logs in this format record the SQL statements that are executed to modify data entries.
  - Row: row-based replication. Binary logs in this format record only modification details about data entries in rows.
  - Mixed: replication in mixed mode. This mode combines the statement and row formats. In most cases, binary logs in the statement format are used to record the SQL statements that are executed to modify data entries, such as functions. If data replication from the primary database to secondary databases cannot be implemented by using binary logs in this format, switch to the row format. MySQL determines which format to use based on each SQL statement that is executed.

  For more information about how to enable the binary logging feature, see Enable the binary logging feature.

Context

Workspaces in standard mode allow you to isolate data sources. You can separately add data sources for the development and production environments to isolate the data sources. This keeps your data secure. For more information, see Isolate connections between the development and production environments.

Limits

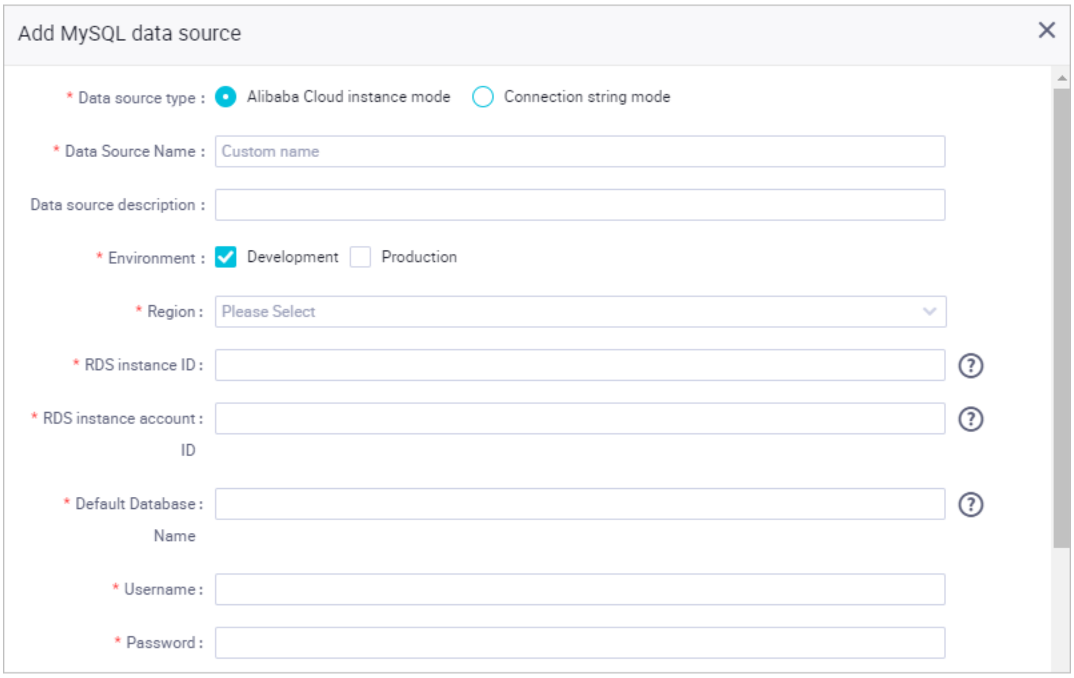
Real-time data synchronization from or to MySQL is performed based on real-time subscription to MySQL binary logs. For a real-time data synchronization node that uses a MySQL data source, you can use an ApsaraDB RDS for MySQL data source whose MySQL version is `V5.x` or `V8.x`. PolarDB for MySQL is not supported.

Add a MySQL data source

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **MySQL** in the Relational Database section.
4. In the **Add MySQL data source** dialog box, configure the parameters.
   i. Configure basic information for the MySQL data source.

      You can use one of the following modes to add a MySQL data source: **Alibaba Cloud Instance Mode** and **Connection String Mode**.

- The following table describes the parameters you can configure when you add a MySQL data source in **Alibaba Cloud instance mode**.


Add MySQL data source

* Data source type : ● Alibaba Cloud instance mode ○ Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Region : Please Select

* RDS instance ID :

* RDS instance account ID :

* Default Database Name :

* Username :

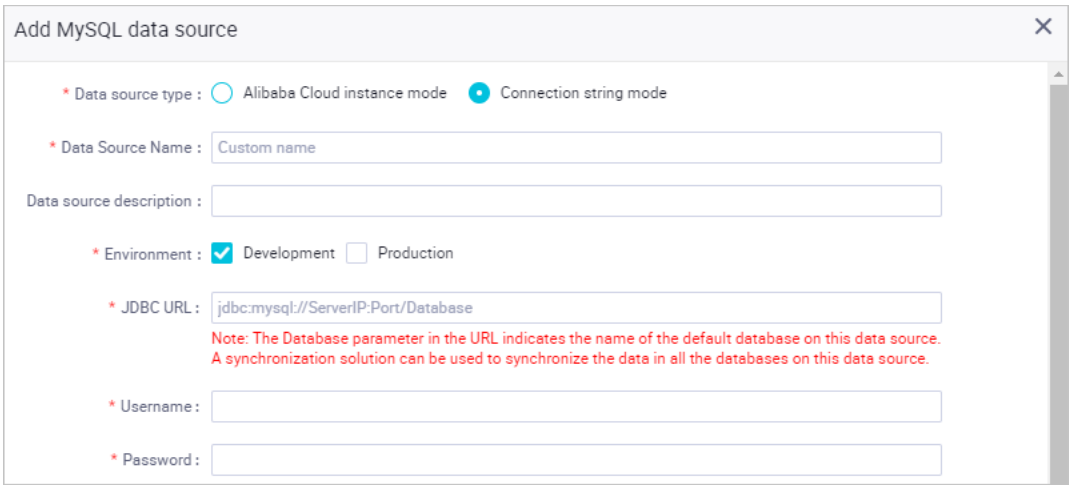
* Password :

Parameter	Description
Data Source Type	The mode in which the data source is added. Set this parameter to **Alibaba Cloud Instance Mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data Source Description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Region	The region where the data source resides.
RDS Instance ID	The ID of the ApsaraDB RDS for MySQL instance. You can log on to the ApsaraDB RDS console to obtain the ID.
RDS Instance Account ID	The ID of the Alibaba Cloud account that is used to purchase the ApsaraDB RDS for MySQL instance. You can use the Alibaba Cloud account to log on to the DataWorks console, move the pointer over the profile picture in the upper-right corner, and then view the ID of the account.

Parameter	Description
Default Database Name	The name of the default ApsaraDB RDS for MySQL database. The following descriptions provide instructions for you to configure a data synchronization node or solution that uses a MySQL data source: ■ When you configure a database-level real-time or batch data synchronization node or solution that uses a MySQL data source, you can select one or more databases on which you have access permissions in the ApsaraDB for MySQL instance. ■ If you select multiple databases when you configure a batch synchronization node, you must add a data source for each database.
Username	The username used to log on to the ApsaraDB RDS for MySQL database.
Password	The password that is used to log on to the ApsaraDB RDS for MySQL database. Do not use at signs (@) in the password.

Note You must add the IP addresses or CIDR blocks that you use to connect to the ApsaraDB RDS for MySQL instance to a whitelist of the instance. For more information, see Configure a whitelist.

- The following table describes the parameters you can configure when you add a MySQL data source in **connection string mode**.



Add MySQL data source

* Data source type : ○ Alibaba Cloud instance mode ● Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* JDBC URL : jdbc:mysql://ServerIP:Port/Database

Note: The Database parameter in the URL indicates the name of the default database on this data source. A synchronization solution can be used to synchronize the data in all the databases on this data source.

* Username :

* Password :

Parameter	Description
Data Source Type	The mode in which the data source is added. Set this parameter to **Connection String Mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data Source Description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database. Specify a value for this parameter in the format of `jdbc:mysql://Server IP address:Port number/Database`. The Database parameter in the URL specifies the name of the default database in this data source. A data synchronization solution or node can be used to synchronize the data in all the databases on which you have access permissions in this data source. **Note** If you use this data source in a batch data synchronization node, the related reader or writer accesses the specified database in the data source by default. If you use this data source in a data synchronization solution, Data Integration accesses all the databases in the data source.
Username	The username that is used to log on to the ApsaraDB RDS for MySQL database.
Password	The password that is used to log on to the ApsaraDB RDS for MySQL database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

> Note
>
> - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
> - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a MySQL data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure MySQL Reader and MySQL Writer. For more information, see MySQL Reader and MySQL Writer.

5.1.12. Add an Oracle data source

DataWorks provides Oracle Reader and Oracle Writer for you to read data from and write data to Oracle data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for Oracle data sources. This topic describes the network environment and permissions that you must prepare before you add a data source. It also describes how to add an Oracle data source in DataWorks.

Prerequisites

Before you add a data source, make sure that the following operations are performed:

- Prepare the data source: An Oracle data source is created.
- Plan and prepare resources: An exclusive resource group for Data Integration is purchased and configured. For more information, see Plan and configure resources.
- Evaluate and plan the network environment: Before you add the data source to DataWorks, connect the data source to an exclusive resource group for Data Integration based on your business requirements. After you connect the data source to the exclusive resource group for Data Integration, configure network access settings such as a vSwitch and a whitelist.
  - If the data sources and the exclusive resource group for Data Integration reside in the same region and virtual private cloud (VPC), they are automatically connected.
  - If the data sources and the exclusive resource group for Data Integration reside in different network environments, you must connect the data sources and the resource group by using methods such as a VPN gateway.

  For more information about how to configure a whitelist, see Configure whitelists for data sources.
- Prepare an account and authorize the account:

  You must create an account that can be used to access the data sources, read data from the source, and write data to the destination during the data synchronization process.

  For more information, see Create and authorize an account.
- Enable the supplemental logging feature:

  You can configure the supplemental logging feature only in a primary Oracle database. This feature can be enabled for a primary or secondary database. For more information about how to enable the feature, see Enable supplemental logging and switch a redo log file.
- Check the character encoding formats of the database:

  For more information, see Check character encoding formats of the database.
- Check the data types of tables in the database:

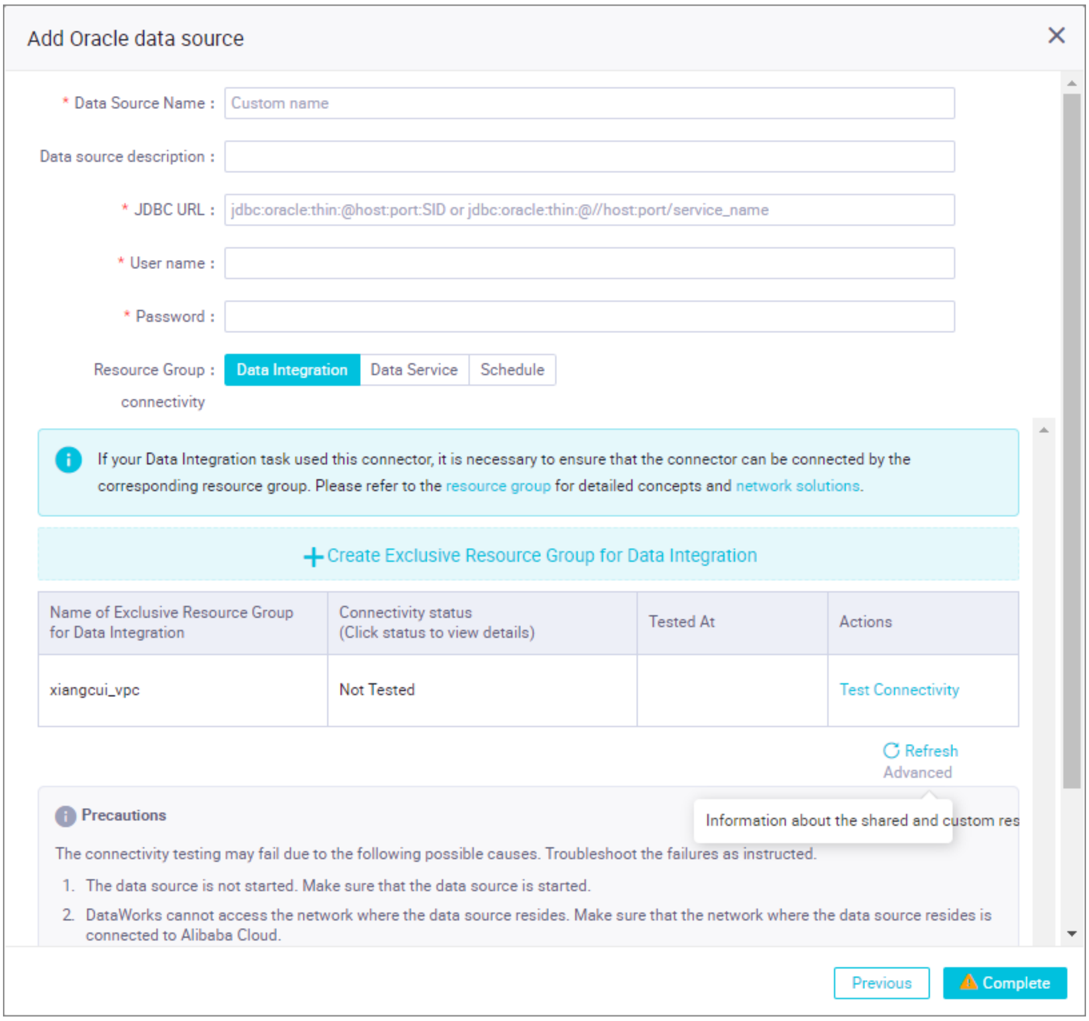
  For more information, see Check the data types of tables in the database.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Add an Oracle data source

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **Oracle** in the Relational Database section.
4. In the **Add Oracle data source** dialog box, configure the parameters.



Add Oracle data source

* Data Source Name : Custom name

Data source description :

* JDBC URL : jdbc:oracle:thin:@host:port:SID or jdbc:oracle:thin:@//host:port/service_name

* User name :

* Password :

Resource Group connectivity : Data Integration | Data Service | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
xiangcui_vpc	Not Tested		Test Connectivity

Refresh
Advanced

Information about the shared and custom res

Precautions

The connectivity testing may fail due to the following possible causes. Troubleshoot the failures as instructed.

1. The data source is not started. Make sure that the data source is started.
2. DataWorks cannot access the network where the data source resides. Make sure that the network where the data source resides is connected to Alibaba Cloud.

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database, in the format of `jdbc:oracle:thin:@ServerIP:Port:Database`.
User name	The username that you use to connect to the database.

Parameter	Description
Password	The password that you use to connect to the database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   Note

   - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.
7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add an Oracle data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure Oracle Reader and Oracle Writer. For more information, see Oracle Reader and Oracle Writer.

5.1.13. Add an OSS data source

Alibaba Cloud Object Storage Service (OSS) is a secure and reliable storage service that allows you to store large volumes of data.

Context

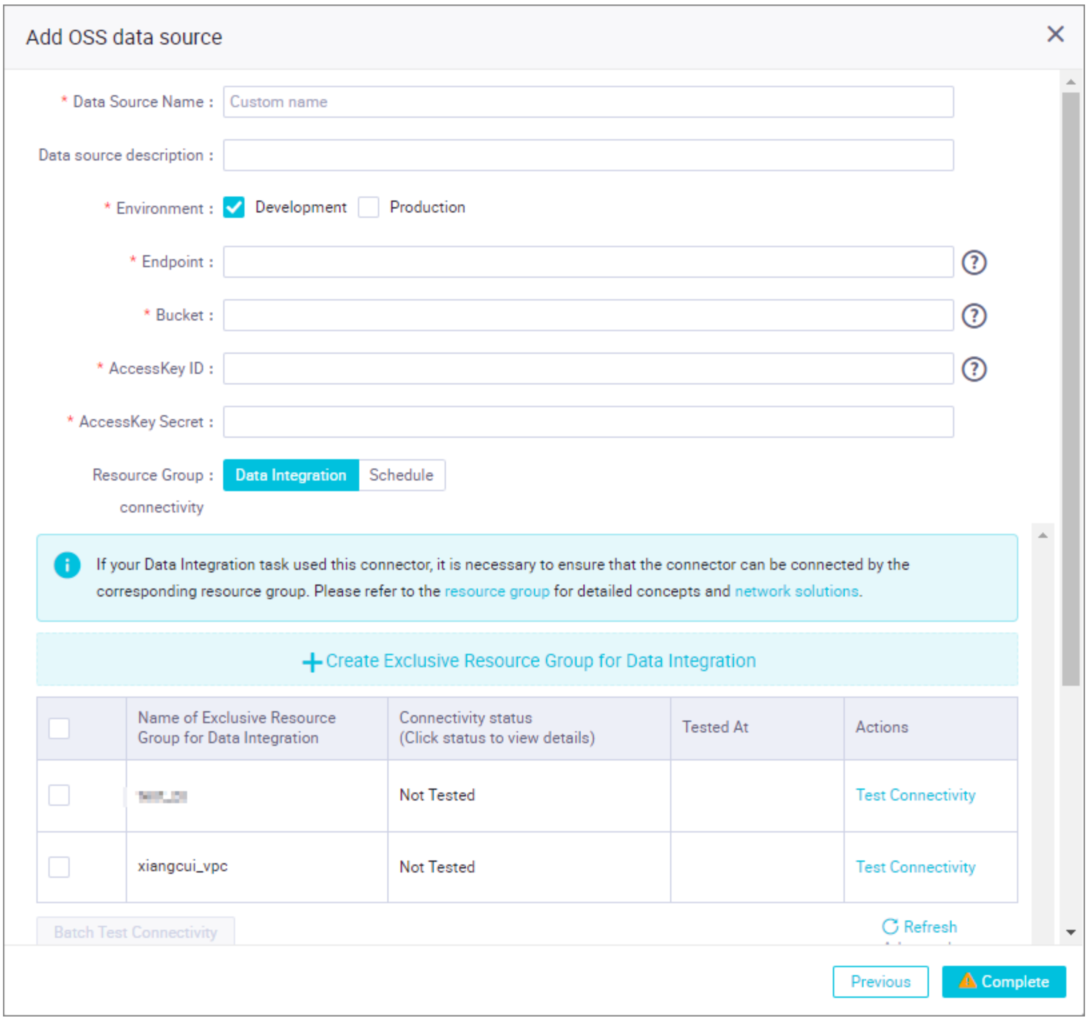
Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

For more information about OSS, see What is OSS?

For more information about OSS SDK for Java, see Aliyun OSS Java SDK.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **OSS** in the Semi-structuredstorage section.
4. In the **Add OSS data source** dialog box, configure the parameters.



Add OSS data source

* Data Source Name :	Custom name
Data source description :	
* Environment :	☑ Development □ Production
* Endpoint :	
* Bucket :	
* AccessKey ID :	
* AccessKey Secret :	
Resource Group connectivity :	Data Integration Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+ Create Exclusive Resource Group for Data Integration

□	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
□	[redacted]	Not Tested		Test Connectivity
□	xiangcui_vpc	Not Tested		Test Connectivity

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Endpoint	The endpoint of OSS. Specify this parameter in the format similar to `http://oss.aliyuncs.com`. The endpoint of OSS varies based on the region. **Note** If you add a bucket name before the endpoint of OSS, the data source can pass the connectivity test but data synchronization will fail. Example of adding a bucket name before the endpoint of OSS: `http://xxx.oss.aliyuncs.com`.

Parameter	Description
Bucket	The name of the OSS bucket. A bucket is a storage space that serves as a container for storing objects. You can create one or more buckets and add one or more objects to each bucket. During data synchronization, DataWorks can search for objects only in the bucket that is specified by this parameter.
AccessKey ID	The AccessKey ID of the account that you can use to connect to the OSS bucket. You can view the AccessKey ID on the Security Management page.
AceessKey Secret	The AccessKey secret of the account that you can use to connect to the OSS bucket.

> **Notice** If data in OSS is stored as CSV files, the data must comply with the standard CSV format. For example, if the data in a column of a CSV file is enclosed in a pair of single quotation marks ('), you must replace this pair of single quotation marks with a pair of double quotation marks ("). Otherwise, the data in the CSV file may be incorrectly parsed.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > **Note**
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add an OSS data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure OSS Reader and OSS Writer. For more information, see OSS Reader and OSS Writer.

5.1.14. Add a Tablestore data source

Tablestore is a NoSQL database service that is built on top of the Apsara distributed operating system. Tablestore allows you to store and access large volumes of structured data in real time.

Context

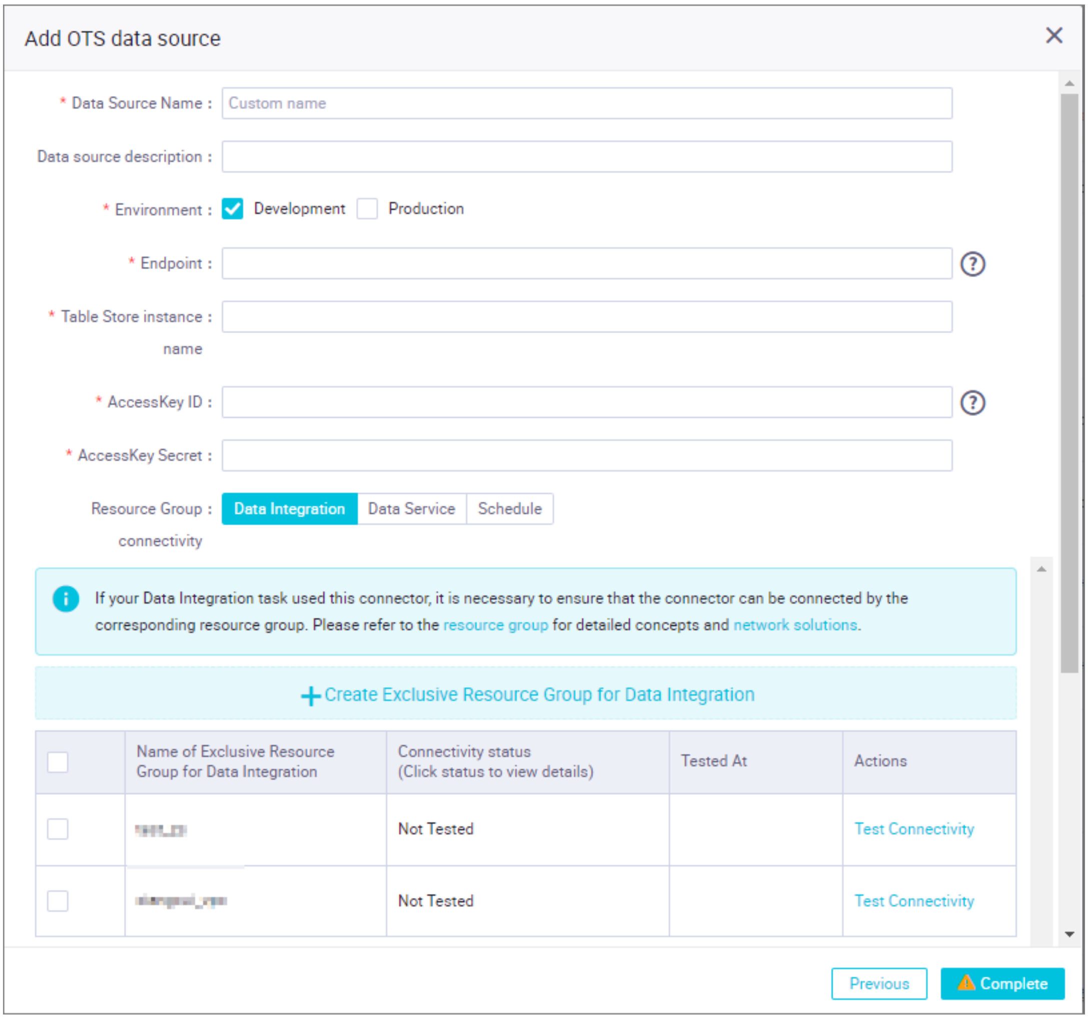
For more information about Tablestore, see What is Tablestore?.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.

iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.

2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **OTS** in the NoSQL section.
4. In the **Add OTS data source** dialog box, configure the parameters.



Add OTS data source

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Endpoint :

* Table Store instance name :

* AccessKey ID :

* AccessKey Secret :

Resource Group connectivity : Data Integration | Data Service | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

□	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
□	[redacted]	Not Tested		Test Connectivity
□	[redacted]	Not Tested		Test Connectivity

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Endpoint	The endpoint of Tablestore.
Table Store instance name	The name of the Tablestore instance.

Parameter	Description
AccessKey ID	The AccessKey ID of the account that you can use to connect to the Tablestore instance. You can view the AccessKey ID on the Security Management page.
AccessKey Secret	The AccessKey secret of the account that you can use to connect to the Tablestore instance.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > Note
   >
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a Tablestore data source. You can proceed to the next tutorial. In the next tutorial, you will learn how to configure Tablestore Reader. For more information, see Tablestore Reader.

5.1.15. Add a PostgreSQL data source

DataWorks provides PostgreSQL Reader and PostgreSQL Writer for you to read data from and write data to PostgreSQL data sources. You can use the codeless user interface (UI) or code editor to configure sync nodes for PostgreSQL data sources.

Context

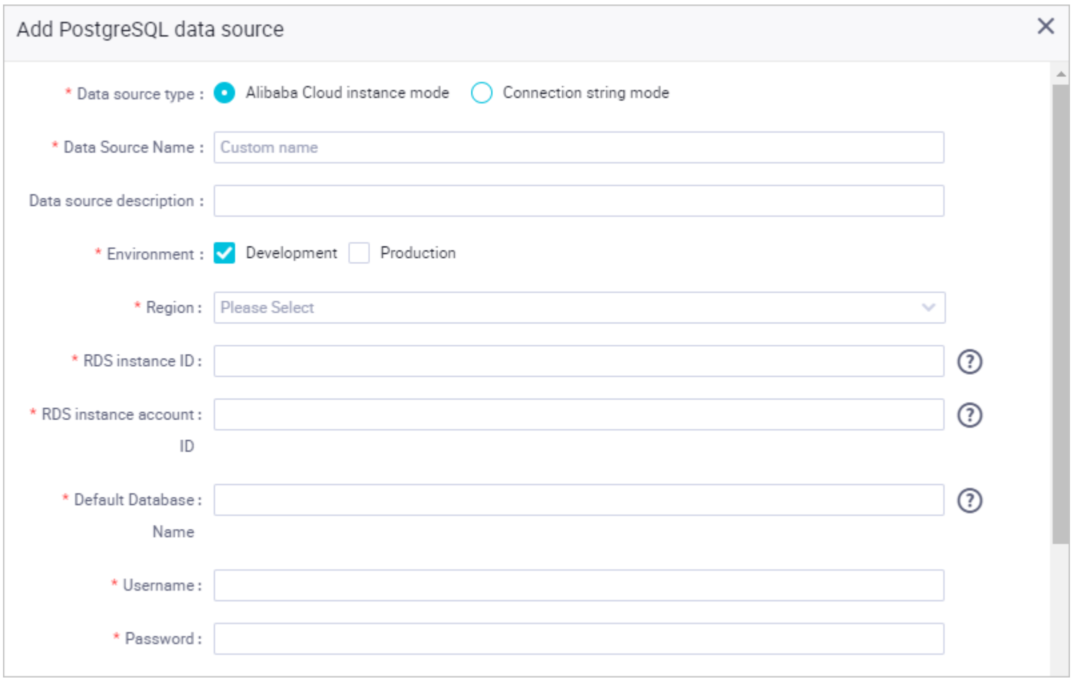
Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **PostgreSQL** in the Relational Database section.
4. In the **Add PostgreSQL data source** dialog box, set the parameters as required.
   i. Configure basic information for the PostgreSQL data source.

      You can use one of the following modes to add a PostgreSQL data source: **Alibaba Cloud instance mode** and **Connection string mode**.

- The following table describes the parameters you can set if you add a PostgreSQL data source by the using **Alibaba Cloud instance mode**.


Add PostgreSQL data source

* Data source type : ● Alibaba Cloud instance mode ○ Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Region : Please Select

* RDS instance ID :

* RDS instance account ID :

* Default Database Name :

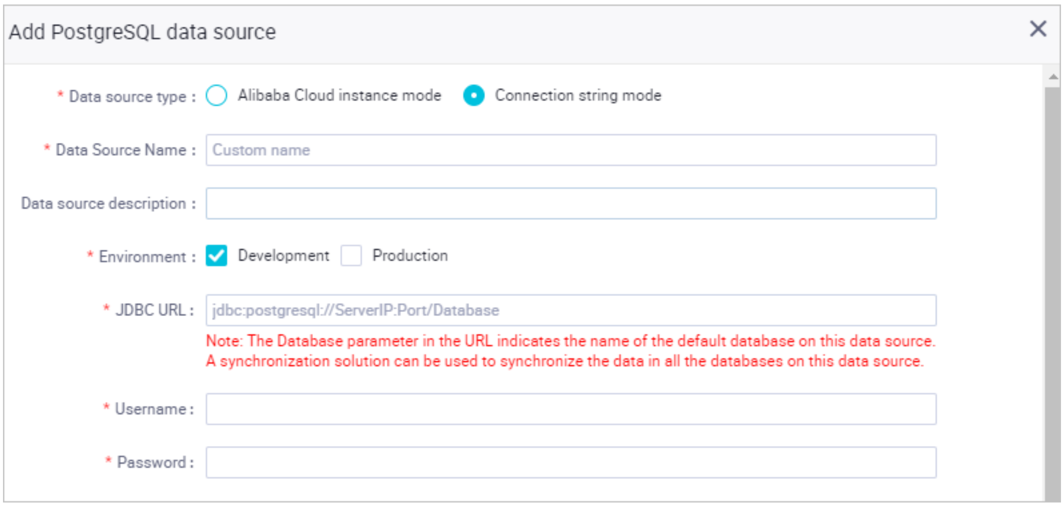
* Username :

* Password :

Parameter	Description
Data source type	The mode in which the data source is added. Set this parameter to **Alibaba Cloud instance mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be up to 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Region	The region in which the data source resides.
RDS instance ID	The ID of the ApsaraDB RDS for PostgreSQL instance. You can log on to the ApsaraDB RDS console to view the ID.
RDS instance account ID	The ID of the Alibaba Cloud account that is used to purchase the ApsaraDB RDS for PostgreSQL instance. You can use the Alibaba Cloud account to log on to the DataWorks console, move the pointer over the profile picture in the upper-right corner, and then view the ID of the account.

Parameter	Description
Database name	The name of the default ApsaraDB RDS for MySQL database. The following descriptions provide instructions for you to configure a data synchronization node or solution that uses a MySQL data source: ■ When you configure a database-level real-time or batch data synchronization node or solution that uses a MySQL data source, you can select one or more databases on which you have access permissions in the ApsaraDB for MySQL instance. ■ If you select multiple databases when you configure a batch synchronization node, you must add a data source for each database.
User name	The username that you use to connect to the database.
Password	The password that you use to connect to the database. Do not use at signs *(@)* in your password.

- The following table describes the parameters you can set if you add a PostgreSQL data source by using the **Connection string mode**.



Add PostgreSQL data source

* Data source type : ○ Alibaba Cloud instance mode ● Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* JDBC URL : jdbc:postgresql://ServerIP:Port/Database

Note: The Database parameter in the URL indicates the name of the default database on this data source. A synchronization solution can be used to synchronize the data in all the databases on this data source.

* Username :

* Password :

Parameter	Description
Data source type	The mode in which the data source is added. Set this parameter to **Connection string mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be up to 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database. Specify this parameter in the format of `jdbc:postgresql://ServerIP:Port/Database`. You must replace Database with the name of the default database. However, when you configure a sync node, you can use all the databases in the ApsaraDB RDS instance.
User name	The username that you use to connect to the database.
Password	The password that you use to connect to the database.

ii.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

> Note
>
> - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
> - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a PostgreSQL data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure PostgreSQL Reader and PostgreSQL Writer. For more information, see PostgreSQL Reader and PostgreSQL Writer.

5.1.16. Add a Redis data source

DataWorks provides Redis Reader and Redis Writer for you to read data from and write data to Redis data sources. You can use the code editor to configure synchronization nodes for Redis data sources.

Context

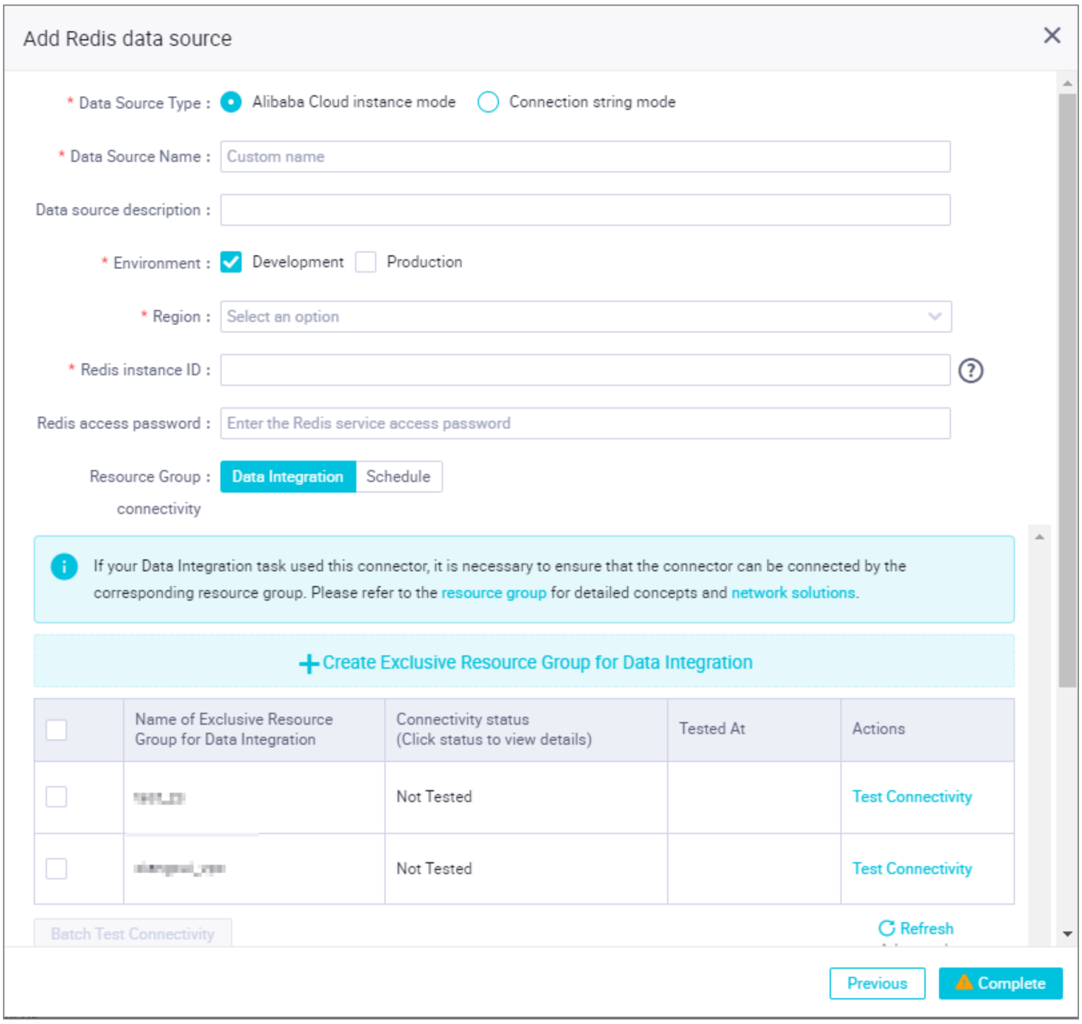
Remote Directory Server (Redis) is a document-based Not only SQL (NoSQL) database that provides persistent and in-memory database services. Redis provides high read and write performance and flexible capacity to meet changing business requirements based on the highly reliable two-node hot standby architecture and the cluster architecture that can be seamlessly scaled.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **Redis** in the NoSQL section.
4. In the **Add Redis data source** dialog box, configure the parameters.

   You can use one of the following modes to add a Redis data source: **Alibaba Cloud instance mode** and **Connection string mode**.

   - The following table describes the parameters you must configure if you add a Redis data source by using the Alibaba Cloud instance mode.



Add Redis data source

* Data Source Type : ● Alibaba Cloud instance mode ○ Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Region : Select an option

* Redis instance ID :

Redis access password : Enter the Redis service access password

Resource Group connectivity : Data Integration | Schedule

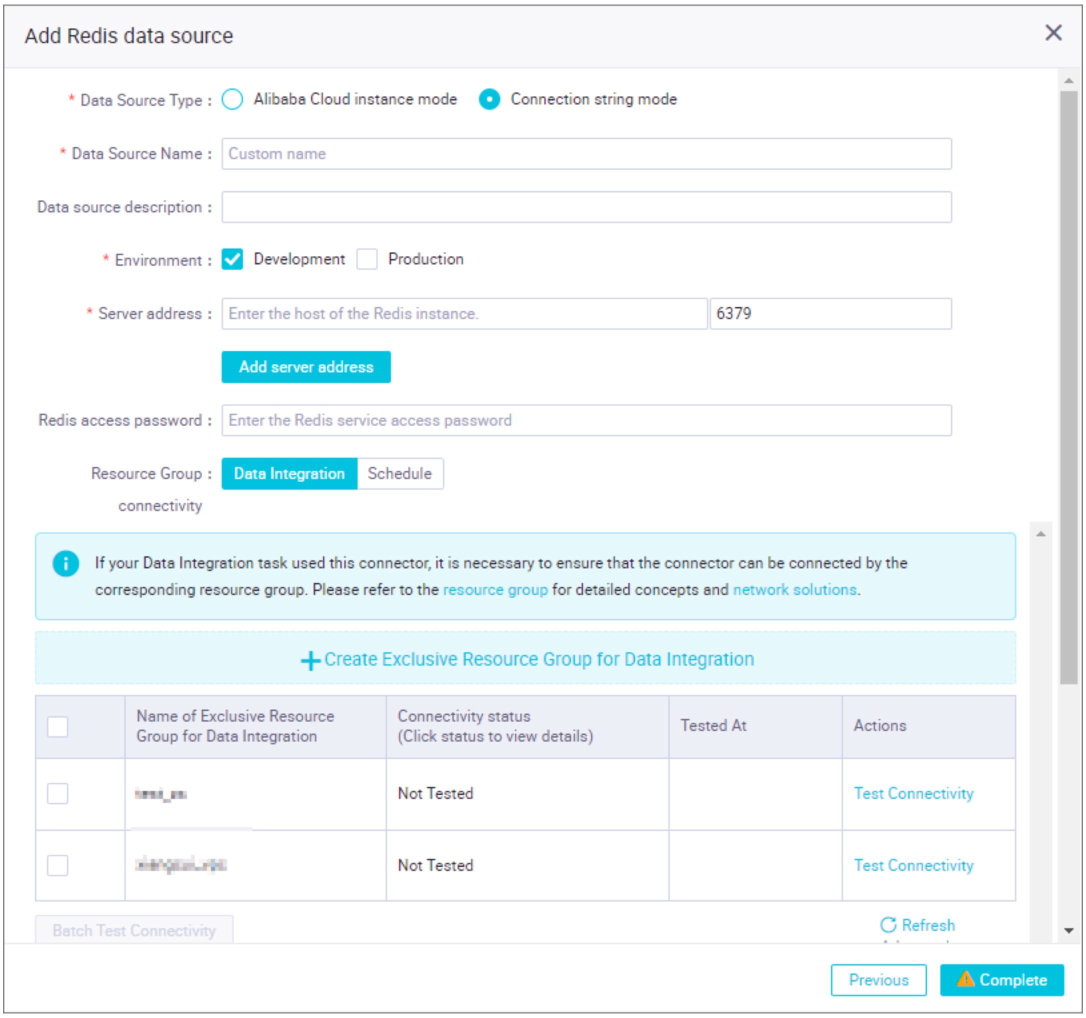
If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

□	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
□	[redacted]	Not Tested		Test Connectivity
□	[redacted]	Not Tested		Test Connectivity

Parameter	Description
Data source type	The type of the data source. Set this parameter to **Alibaba Cloud instance mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only the workspace is in standard mode.
Region	The region where the data source resides.
Redis instance ID	The ID of the ApsaraDB for Redis instance. You can log on to the ApsaraDB for Redis console to view the ID.
Redis access password	The password that is used to connect to the ApsaraDB for Redis instance. Leave it empty if no password is required.

- The following table describes the parameters you must configure if you add a Redis data source by using the connection string mode.



Add Redis data source ×

* Data Source Type : ○ Alibaba Cloud instance mode ● Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Server address : Enter the host of the Redis instance. 6379

Add server address

Redis access password : Enter the Redis service access password

Resource Group connectivity : Data Integration Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

□	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
□	[redacted]	Not Tested		Test Connectivity
□	[redacted]	Not Tested		Test Connectivity

Batch Test Connectivity

Refresh

Previous

Complete

Parameter	Description
Data source type	The type of the data source. Set this parameter to **Connection string mode**. If you select this mode, you must use custom resource groups for scheduling to run synchronization nodes. For more information, see **Create a custom resource group for scheduling**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Server address	The server address. Specify this parameter in the format of `Host address:Port number`.
Add server address	Click **Add server address** to add a server address in the format of `Host address:Port number`.

Parameter	Description
Redis access password	The password that is used to connect to the ApsaraDB for Redis instance.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > Note
   >
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a Redis data source. You can proceed to the next tutorial. In the next tutorial, you will learn how to configure Redis Writer. For more information, see Redis Writer.

5.1.17. Add a HybridDB for MySQL data source

DataWorks provides HybridDB for MySQL Reader and HybridDB for MySQL Writer for you to read data from and write data to HybridDB for MySQL data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for HybridDB for MySQL data sources.

Context

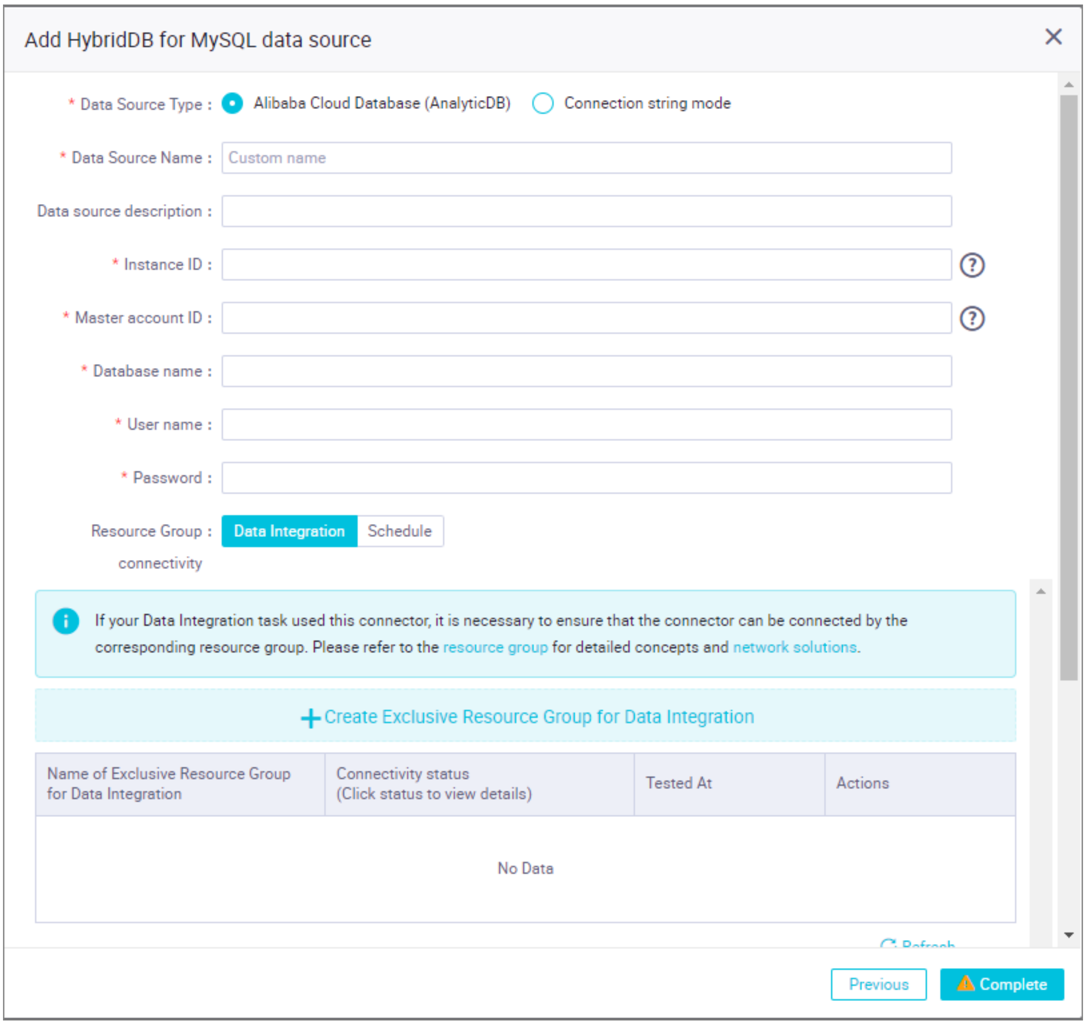
Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Procedure

1. Go to the **Data Source** page.
   1. Log on to the DataWorks console.
   2. In the left-side navigation pane, click **Workspaces**.
   3. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   4. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **HybridDB for MySQL** in the Relational Database section.
4. In the **Add HybridDB for MySQL data source** dialog box, configure the parameters.

   You can use one of the following modes to add a HybridDB for MySQL data source: **Alibaba Cloud Database (AnalyticDB)** and **Connection string mode**.

   - The following table describes the parameters you must configure if you add a HybridDB for MySQL data source by using the Alibaba Cloud database (AnalyticDB) mode.



Add HybridDB for MySQL data source

* Data Source Type : ● Alibaba Cloud Database (AnalyticDB) ○ Connection string mode

* Data Source Name : Custom name

Data source description :

* Instance ID :

* Master account ID :

* Database name :

* User name :

* Password :

Resource Group connectivity : Data Integration Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

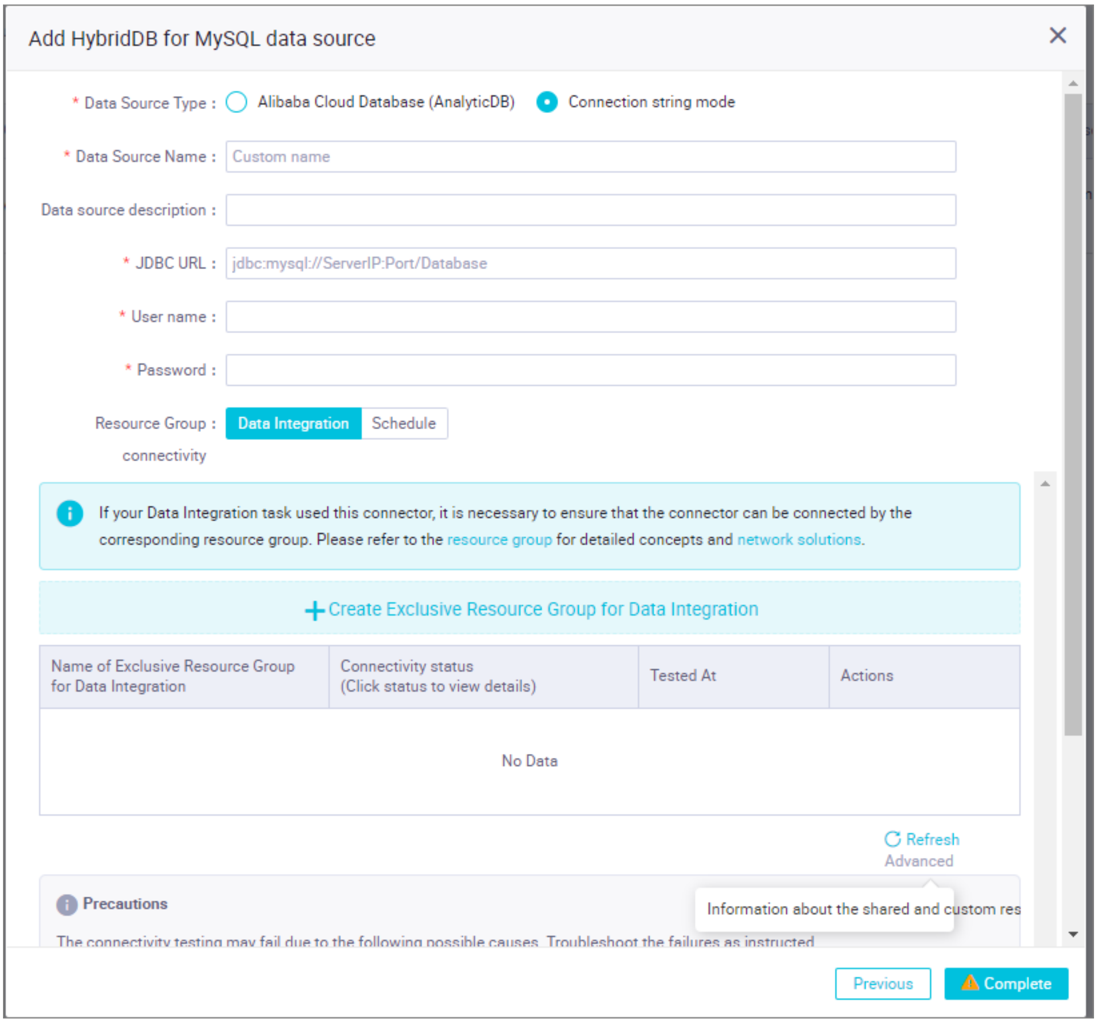
Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
No Data			

Refresh

Previous Complete

Parameter	Description
Data source type	The type of the data source. Set this parameter to **Alibaba Cloud Database (AnalyticDB)**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. ⓘ Note This parameter is displayed only when the workspace is in standard mode.
Instance ID	The ID of the HybridDB for MySQL instance. You can log on to the HybridDB for MySQL console to view the ID.
Master account ID	The ID of the Alibaba Cloud account that is used to purchase the HybridDB for MySQL instance. You can view the ID of the account on the **Security Settings** page.
User name	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

- The following table describes the parameters you must configure if you add a HybridDB for MySQL data source by using the connection string mode.



Add HybridDB for MySQL data source ×

* Data Source Type : ○ Alibaba Cloud Database (AnalyticDB) ● Connection string mode

* Data Source Name : Custom name

Data source description :

* JDBC URL : jdbc:mysql://ServerIP:Port/Database

* User name :

* Password :

Resource Group connectivity : Data Integration Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+ Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
No Data			

Refresh

Advanced

Precautions

Information about the shared and custom res

The connectivity testing may fail due to the following possible causes. Troubleshoot the failures as instructed

Parameter	Description
Data source type	The type of the data source. Set this parameter to **Connection string mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. Note This parameter is displayed only when the workspace is in standard mode.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database. Specify this parameter in the format of `jdbc:mysql://Server IP address:Port number/Database`.
User name	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > Note
   >
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a HybridDB for MySQL data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure HybridDB for MySQL Reader and HybridDB for MySQL Writer. For more information, see HybridDB for MySQL Reader and HybridDB for MySQL Writer.

5.1.18. Add an AnalyticDB for PostgreSQL data source

DataWorks provides AnalyticDB for PostgreSQL Reader and AnalyticDB for PostgreSQL Writer for you to read data from and write data to AnalyticDB for PostgreSQL data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for AnalyticDB for PostgreSQL data sources.

Context

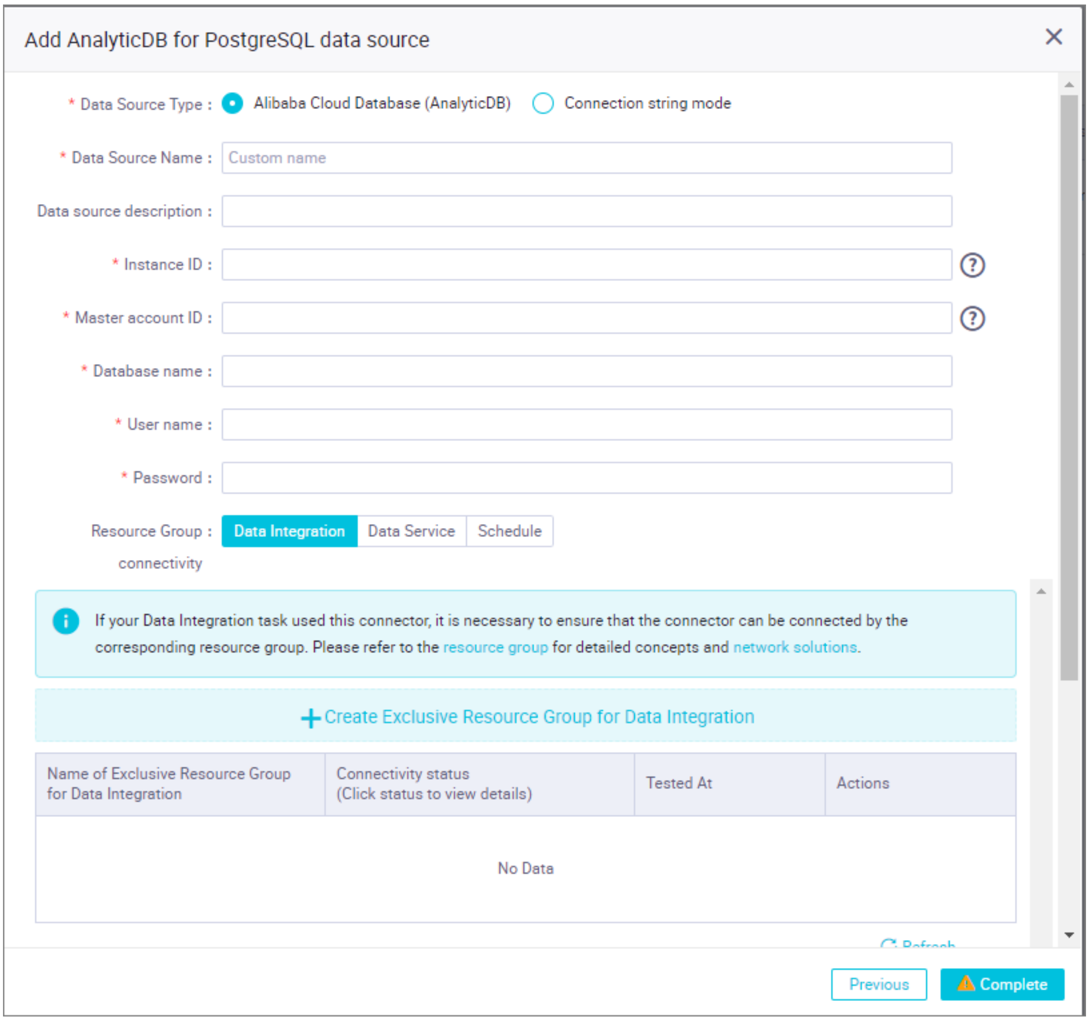
Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **AnalyticDB for PostgreSQL** in the Relational Database section.
4. In the **Add AnalyticDB for PostgreSQL data source** dialog box, configure the parameters.

   You can use one of the following modes to add an AnalyticDB for PostgreSQL data source: **Alibaba Cloud Database (AnalyticDB)** and **Connection string mode**.

   - The following table describes the parameters you must configure if you add an AnalyticDB for PostgreSQL data source by using the Alibaba Cloud database (AnalyticDB) mode.


Add AnalyticDB for PostgreSQL data source

* Data Source Type : ● Alibaba Cloud Database (AnalyticDB) ○ Connection string mode

* Data Source Name : Custom name

Data source description :

* Instance ID :

* Master account ID :

* Database name :

* User name :

* Password :

Resource Group connectivity : Data Integration | Data Service | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

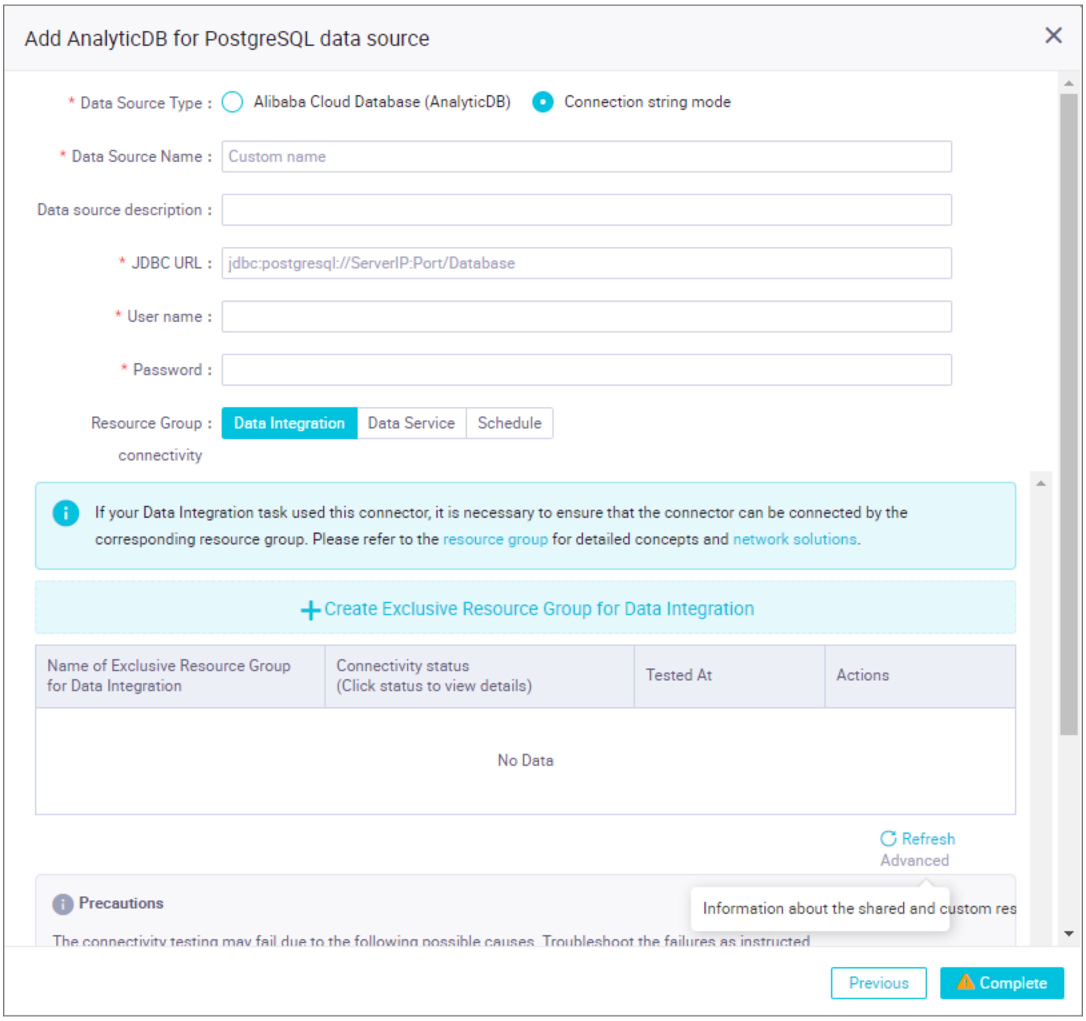
+ Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
No Data			

Parameter	Description
Data source type	The type of the data source. Set this parameter to **Alibaba Cloud Database (AnalyticDB)**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Instance ID	The ID of the AnalyticDB for PostgreSQL instance. You can log on to the AnalyticDB for PostgreSQL console to view the ID.
Master account ID	The ID of the Alibaba Cloud account that is used to purchase the AnalyticDB for PostgreSQL instance. You can view the ID of the account on the **Security Settings** page.
Database name	The name of the AnalyticDB for PostgreSQL database.

Parameter	Description
User name	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

- The following table describes the parameters you must configure if you add an AnalyticDB for PostgreSQL data source by using the connection string mode.



Add AnalyticDB for PostgreSQL data source

* Data Source Type : ○ Alibaba Cloud Database (AnalyticDB) ● Connection string mode

* Data Source Name : Custom name

Data source description :

* JDBC URL : jdbc:postgresql://ServerIP:Port/Database

* User name :

* Password :

Resource Group connectivity : Data Integration | Data Service | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
No Data			

Precautions

Information about the shared and custom res

The connectivity testing may fail due to the following possible causes. Troubleshoot the failures as instructed

Parameter	Description
Data source type	The type of the data source. Set this parameter to **Connection string mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.

Parameter	Description
JDBC URL	The Java Database Connectivity (JDBC) URL of the database. Specify this parameter in the format of `jdbc:postgresql://Server IP address:Port number/Database`.
User name	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > Note
   >
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add an AnalyticDB for PostgreSQL data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure AnalyticDB for PostgreSQL Reader and AnalyticDB for PostgreSQL Writer. For more information, see AnalyticDB for PostgreSQL Reader and AnalyticDB for PostgreSQL Writer.

5.1.19. Add a PolarDB data source

DataWorks provides PolarDB Reader and PolarDB Writer for you to read data from and write data to PolarDB data sources. You can use the codeless user interface (UI) or code editor to configure sync nodes for PolarDB data sources.

Context

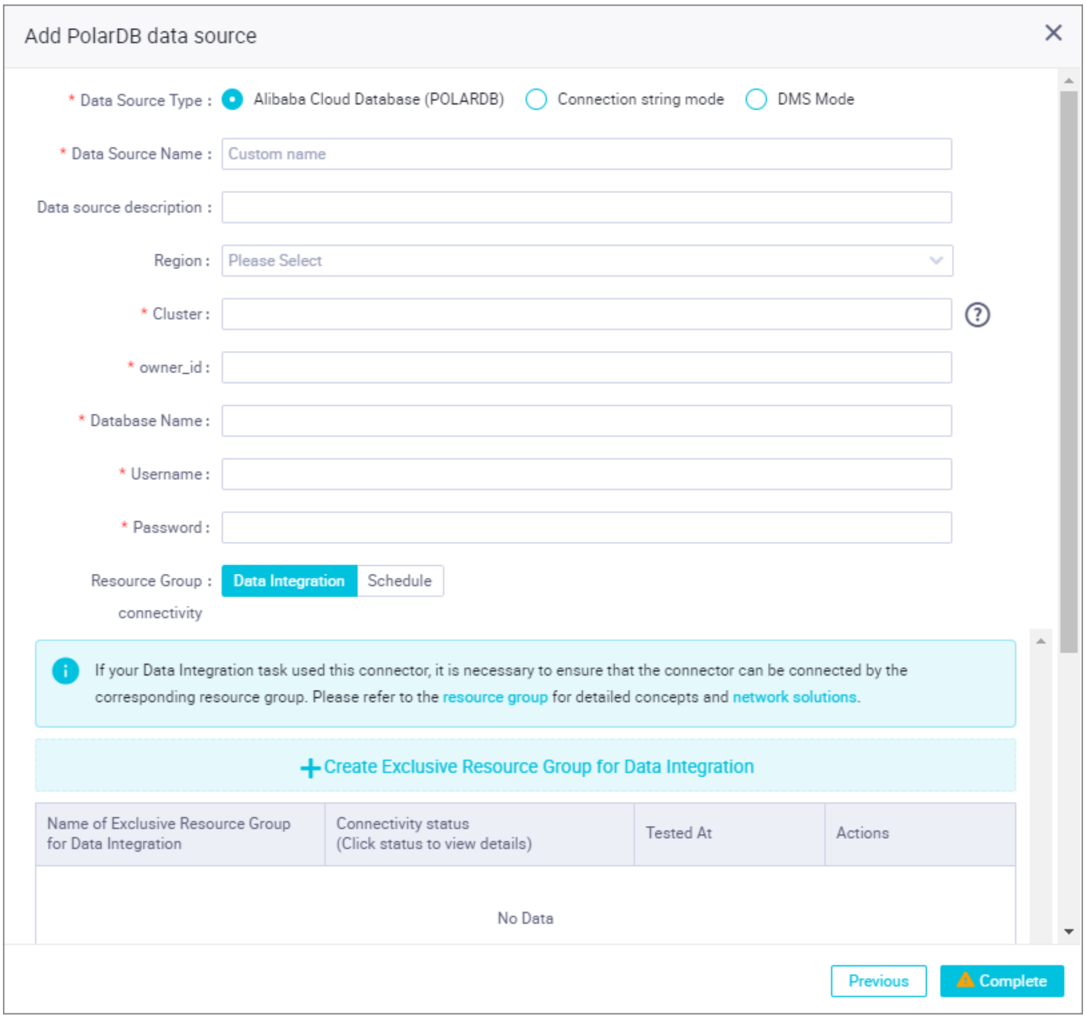
Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source** > **Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **PolarDB** in the Relational Database section.
4. In the **Add PolarDB data source** dialog box, set the parameters as required.

You can use one of the following modes to add a PolarDB data source: **Alibaba Cloud Database (POLARDB)** and **Connection string mode**.

- The following table describes the parameters you can set if you add a **PolarDB** data source by using the **Alibaba Cloud Database (POLARDB)** mode.



Add PolarDB data source

* Data Source Type : ● Alibaba Cloud Database (POLARDB) ○ Connection string mode ○ DMS Mode

* Data Source Name : Custom name

Data source description :

Region : Please Select

* Cluster :

* owner_id :

* Database Name :

* Username :

* Password :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

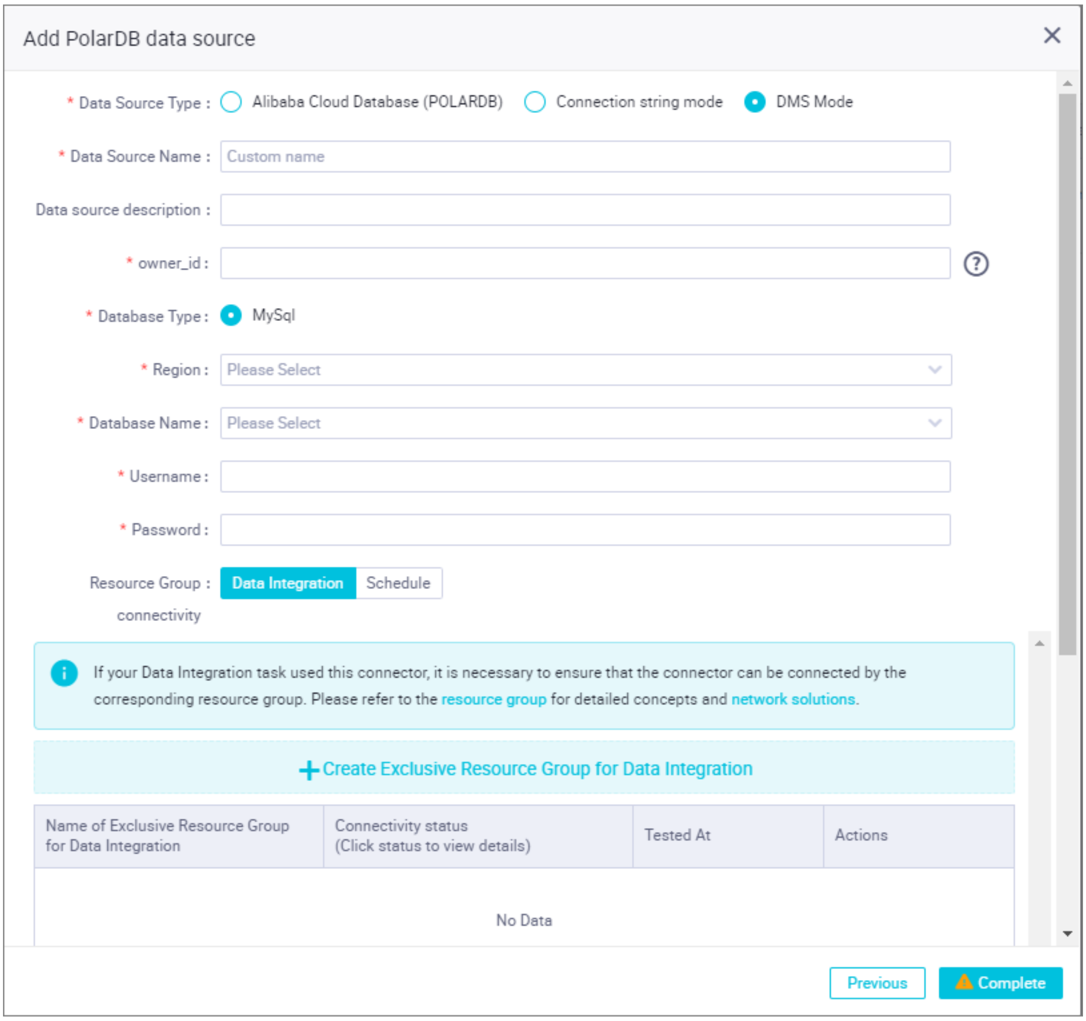
Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions

No Data

Parameter	Description
Data source type	The mode in which the data source is added. Set this parameter to **Alibaba Cloud Database (POLARDB)**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be up to 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Cluster ID	The ID of the PolarDB cluster. You can log on to the PolarDB console to view the ID.

Parameter	Description
POLARDB instance master account ID	The ID of the Alibaba Cloud account that is used to purchase the PolarDB cluster. You can use the account to log on to the DataWorks console, move the pointer over the profile picture in the upper-right corner, and then view the ID of the account.
Database name	The name of the PolarDB database.
User name	The username that you use to connect to the database.
Password	The password that you use to connect to the database.

- The following table describes the parameters you can set if you add a **PolarDB** data source by using the **Connection string mode**.



Add PolarDB data source

* Data Source Type : ○ Alibaba Cloud Database (POLARDB) ○ Connection string mode ● DMS Mode

* Data Source Name : Custom name

Data source description :

* owner_id :

* Database Type : ● MySql

* Region : Please Select

* Database Name : Please Select

* Username :

* Password :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

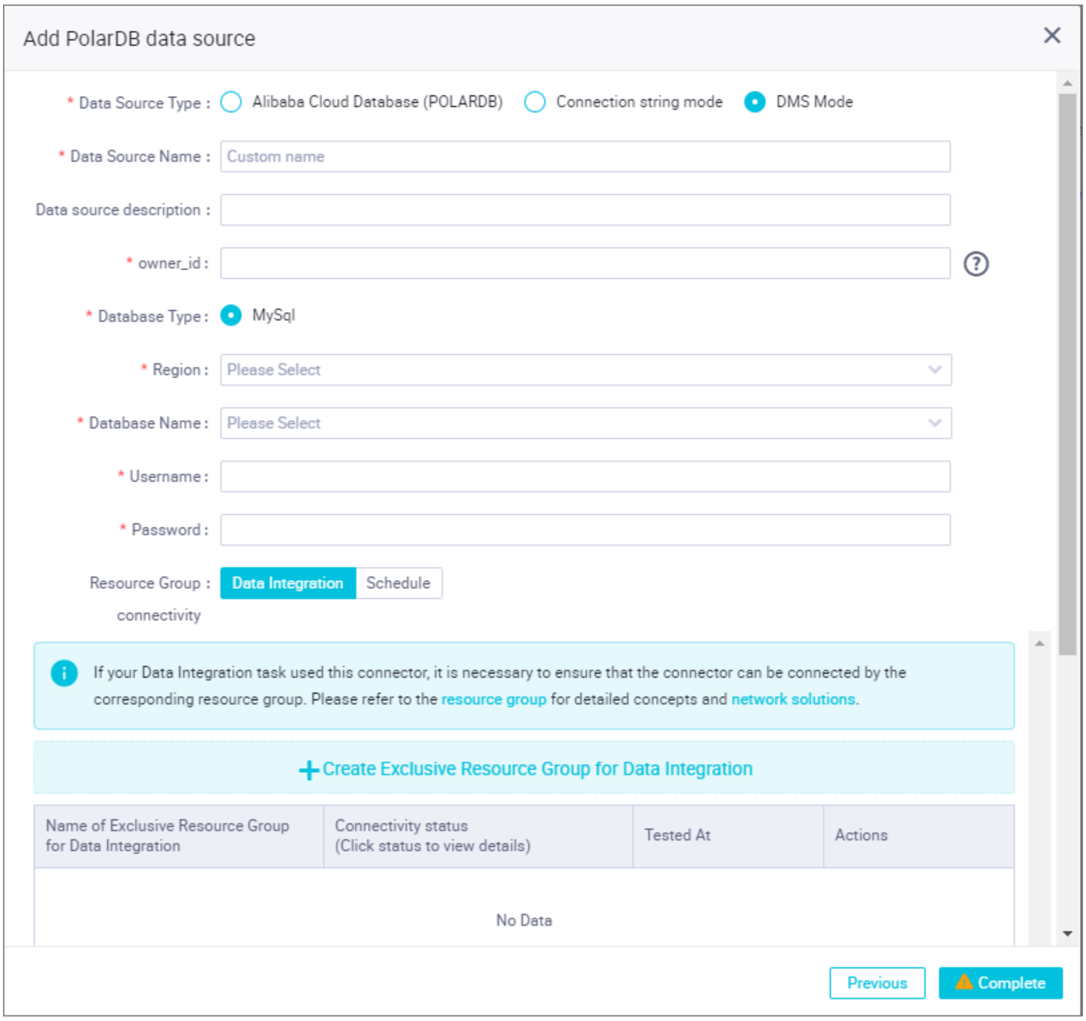
+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
No Data			

Parameter	Description
Data source type	The mode in which the data source is added. Set this parameter to **Connection string mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be up to 80 characters in length.

Parameter	Description
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. Note This parameter is displayed only when the workspace is in standard mode.
Database type	The type of the database. Valid values: **MySql** and **PostgreSql**.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database. Specify this parameter in the format of `jdbc:mysql://ServerIP:Port/Database`.
User name	The username that you use to connect to the database.
Password	The password that you use to connect to the database.

- The following table describes the parameters you must configure if you add a **PolarDB** data source by using the **DMS Mode**.



Add PolarDB data source

* Data Source Type : ○ Alibaba Cloud Database (POLARDB) ○ Connection string mode ● DMS Mode

* Data Source Name : Custom name

Data source description :

* owner_id :

* Database Type : ● MySql

* Region : Please Select

* Database Name : Please Select

* Username :

* Password :

Resource Group connectivity : Data Integration Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
No Data			

Parameter	Description
Data source type	The mode in which the data source is added. Set this parameter to **DMS Mode**.

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be up to 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
owner_id	The ID of the Alibaba Cloud account that is used to purchase the PolarDB cluster. You can use the account to log on to the DataWorks console, move the pointer over the profile picture in the upper-right corner, and then view the ID of the account.
Database type	The type of the database. Default value: **MySql**.
Region	The region in which the data source resides.
Database name	The name of the PolarDB database.
User name	The username that you use to connect to the database.
Password	The password that you use to connect to the database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > Note
   >
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a PolarDB data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure PolarDB Reader and PolarDB Writer. For more information, see PolarDB Reader and PolarDB Writer.

5.1.20. Configure an AnalyticDB for MySQL 3.0 connection

The AnalyticDB MySQL 3.0 data source provides you with a two-way channel for reading and writing AnalyticDB 3.0. You can configure synchronization tasks in wizard mode and script mode.

Context

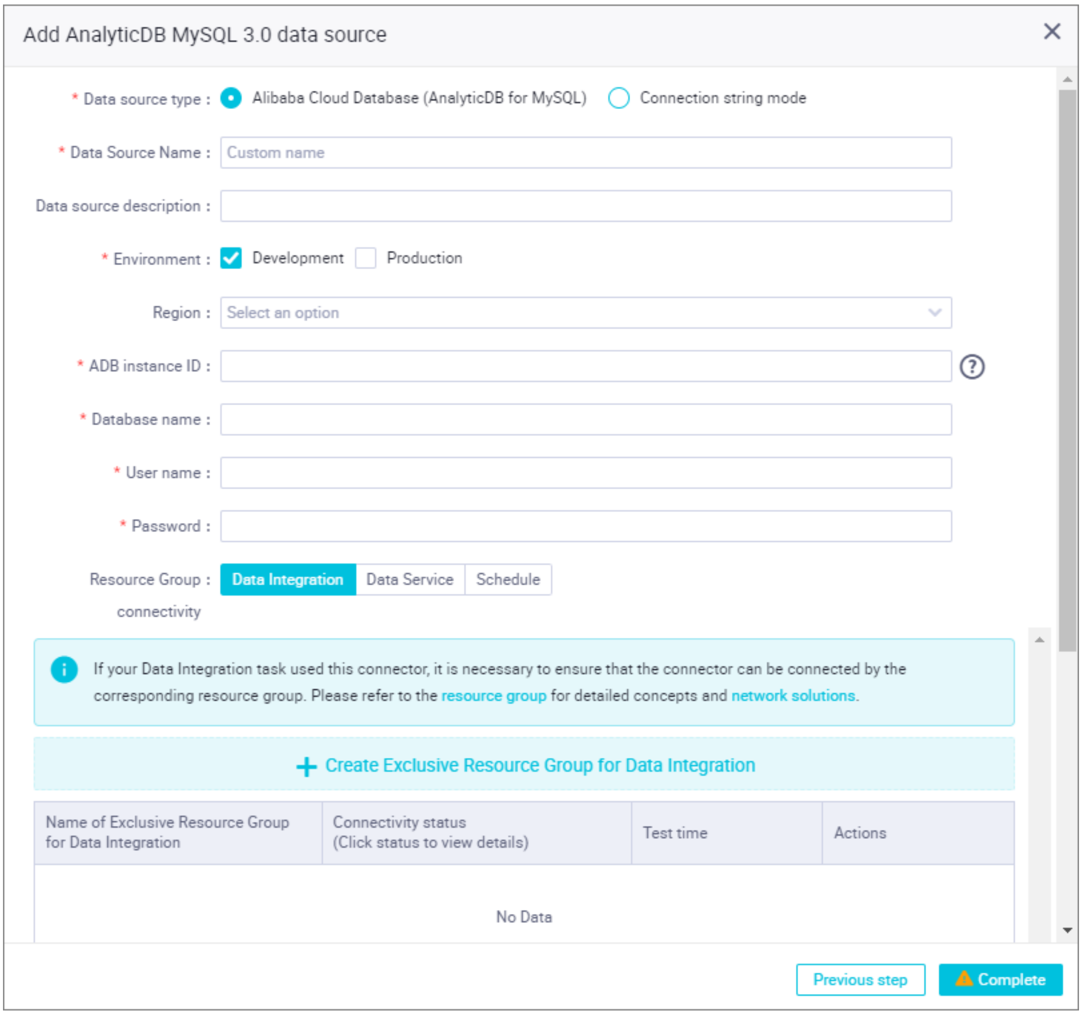
Workspaces in standard mode support the Isolate connections between the development and production environments feature. You can add data sources for the development and production environments separately and isolate the data sources to protect your data security.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add Data Source** dialog box, set Data Source Type to **AnalyticDB MySQL 3.0**.
4. In the **Add AnalyticDB for MySQL 3.0 Data Source** dialog box, set the parameters as required.

   Data source types include **Alibaba Cloud Database (AnalyticDB for MySQL)** and **connection string mode**. You can select data source types based on your needs.

   - The following example shows how to add a data source of the **AnalyticDB MySQL 3.0 > Alibaba Cloud Database (AnalyticDB for MySQL)** type.



Add AnalyticDB MySQL 3.0 data source

* Data source type : ● Alibaba Cloud Database (AnalyticDB for MySQL) ○ Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

Region : Select an option

* ADB instance ID :

* Database name :

* User name :

* Password :

Resource Group connectivity : Data Integration | Data Service | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+ Create Exclusive Resource Group for Data Integration

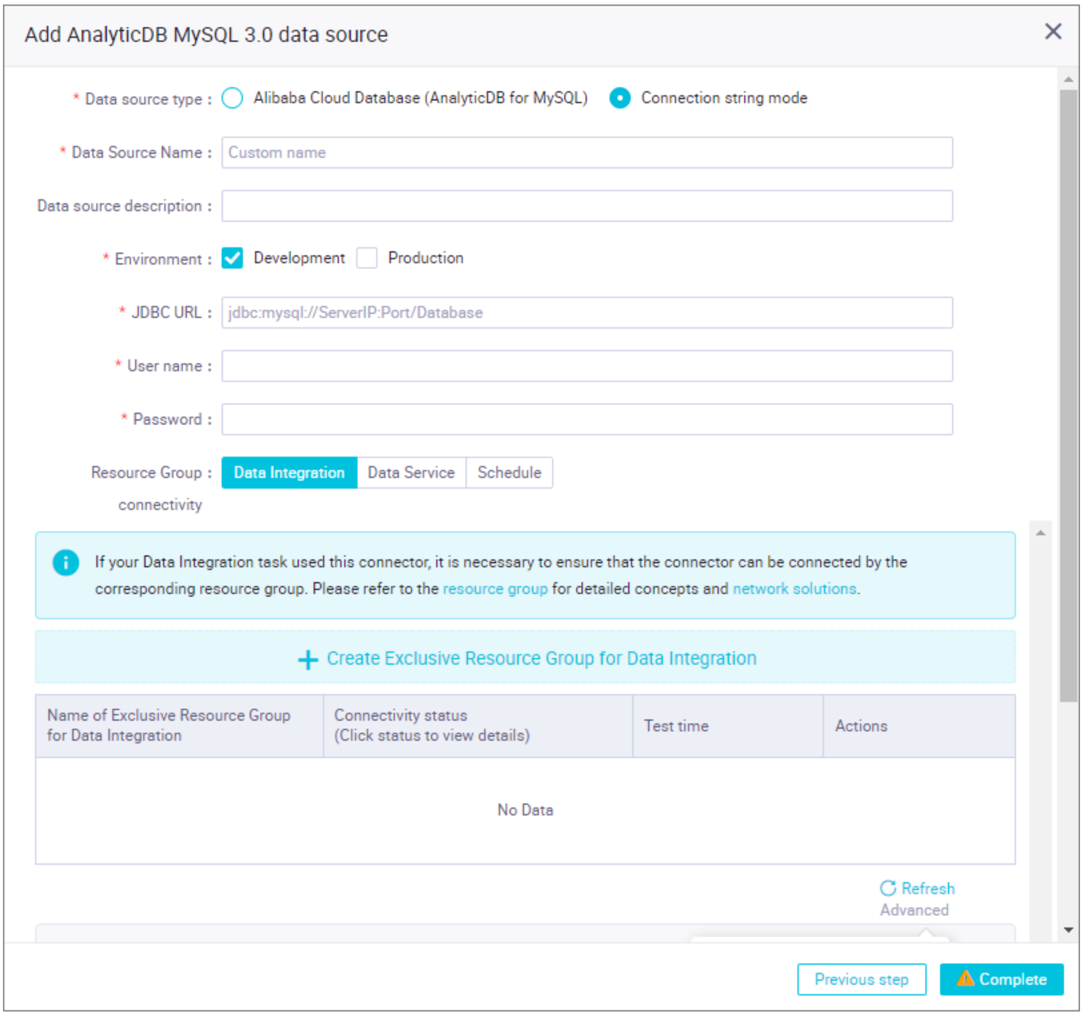
Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Test time	Actions
No Data			

Previous step | Complete

Parameter	Description
Data source type	The data source type is **AnalyticDB MySQL 3.0 > Alibaba Cloud Database (AnalyticDB for MySQL)** .

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Region	The region where the ApsaraDB RDS for PostgreSQL instance resides.
ADB Instance ID	You can go to the AnalyticDB for MySQL 3.0 console to view the instance ID.
Database name	The name of the ApsaraDB RDS for MySQL database.
User name	The username that is used to connect to the ApsaraDB RDS for MySQL database.
Password	The password that is used to connect to the ApsaraDB RDS for MySQL database.

- The following example shows how to add a data source of the **AnalyticDB MySQL 3.0 > connection string mode** type.



Add AnalyticDB MySQL 3.0 data source

* Data source type : ○ Alibaba Cloud Database (AnalyticDB for MySQL) ● Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* JDBC URL : jdbc:mysql://ServerIP:Port/Database

* User name :

* Password :

Resource Group connectivity : Data Integration | Data Service | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+ Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Test time	Actions
No Data			

Parameter	Description
Data source type	**AnalyticDB MySQL 3.0 > connection string mode** is the currently selected data source type.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. Note This parameter is available only when the workspace is in standard mode.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database, in the format of `jdbc:mysql://ServerIP:Port/Database`.
User name	The username that is used to connect to the ApsaraDB RDS for MySQL database.
Password	The password that is used to connect to the ApsaraDB RDS for MySQL database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > Note
   >
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

Now you have learned how to configure the AnalyticDB for MySQL 3.0 data source. You can proceed with the next tutorial. In this tutorial, you will learn how to configure the AnalyticDB for MySQL 3.0 plug-in. For more information, see AnalyticDB MySQL 3.0 Reader and AnalyticDB MySQL 3.0 Writer.

5.1.21. Add a ClickHouse data source

DataWorks provides ClickHouse Reader and ClickHouse Writer for you to read data from and write data to ClickHouse data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for ClickHouse data sources.

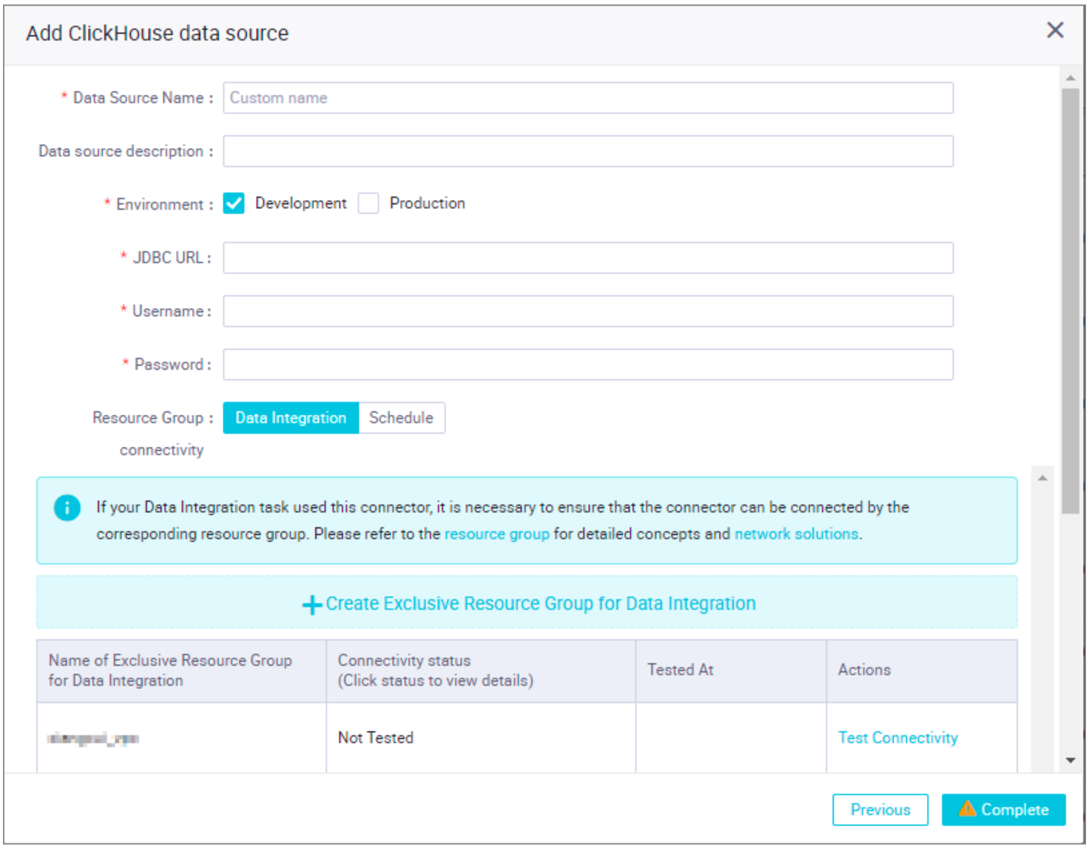
Context

> Note Only ApsaraDB for ClickHouse data sources are supported.

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information about the feature, see Isolate connections between the development and production environments.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **ClickHouse** in the Big Data Storage section.
4. In the **Add ClickHouse data source** dialog box, configure the following parameters.



Add ClickHouse data source

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* JDBC URL :

* Username :

* Password :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
[redacted]	Not Tested		Test Connectivity

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data Source Description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. Note This parameter is displayed only when the workspace is in standard mode.
JDBC URL	The Java Database Connectivity (JDBC) URL of the data source. Specify a value for this parameter in the format of `jdbc:clickhouse://ServerIP:Port/Database`. Note Only HTTP port 8123 is supported.
Username	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

> Note
>
> - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
> - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a ClickHouse data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure ClickHouse Reader and ClickHouse Writer. For more information, see ClickHouse Reader and ClickHouse Writer.

5.1.22. Add a DLA data source

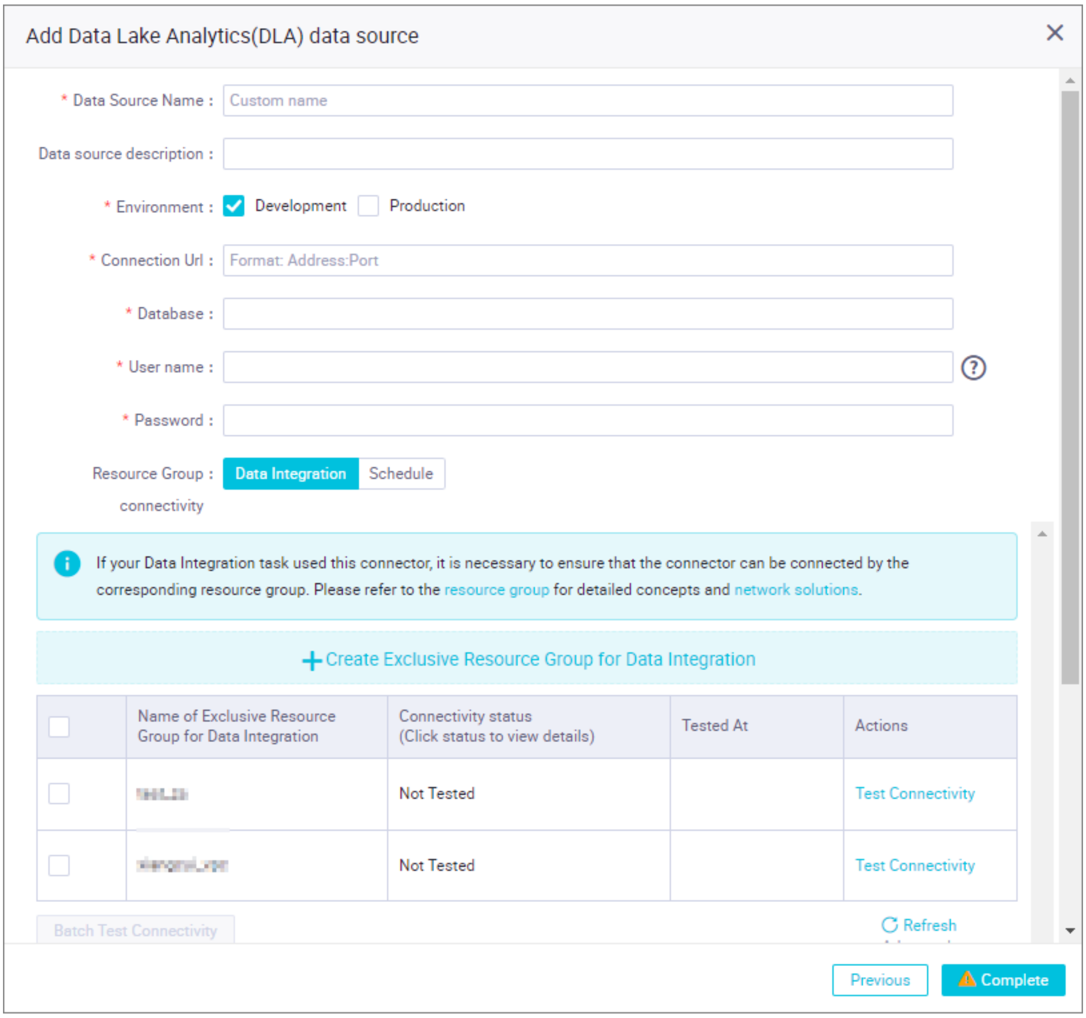
This topic describes how to add a Data Lake Analytics (DLA) data source.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **Data Lake Analytics(DLA)** in the Big Data Storage section.
4. In the **Add Data Lake Analytics(DLA) data source** dialog box, configure the parameters.



Add Data Lake Analytics(DLA) data source

* Data Source Name :	Custom name
Data source description :	
* Environment :	☑ Development □ Production
* Connection Url :	Format: Address:Port
* Database :	
* User name :	
* Password :	
Resource Group connectivity :	Data Integration / Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+ Create Exclusive Resource Group for Data Integration

□	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
□	[redacted]	Not Tested		Test Connectivity
□	[redacted]	Not Tested		Test Connectivity

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Connection Url	The URL of the DLA database. Specify this parameter in the format of `Address:Port number`.
Database	The name of the DLA database.
User name	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

> Note
>
> - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
> - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

5.1.23. Add a MaxCompute data source

DataWorks provides MaxCompute Reader and MaxCompute Writer for you to read data from and write data to MaxCompute data sources.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

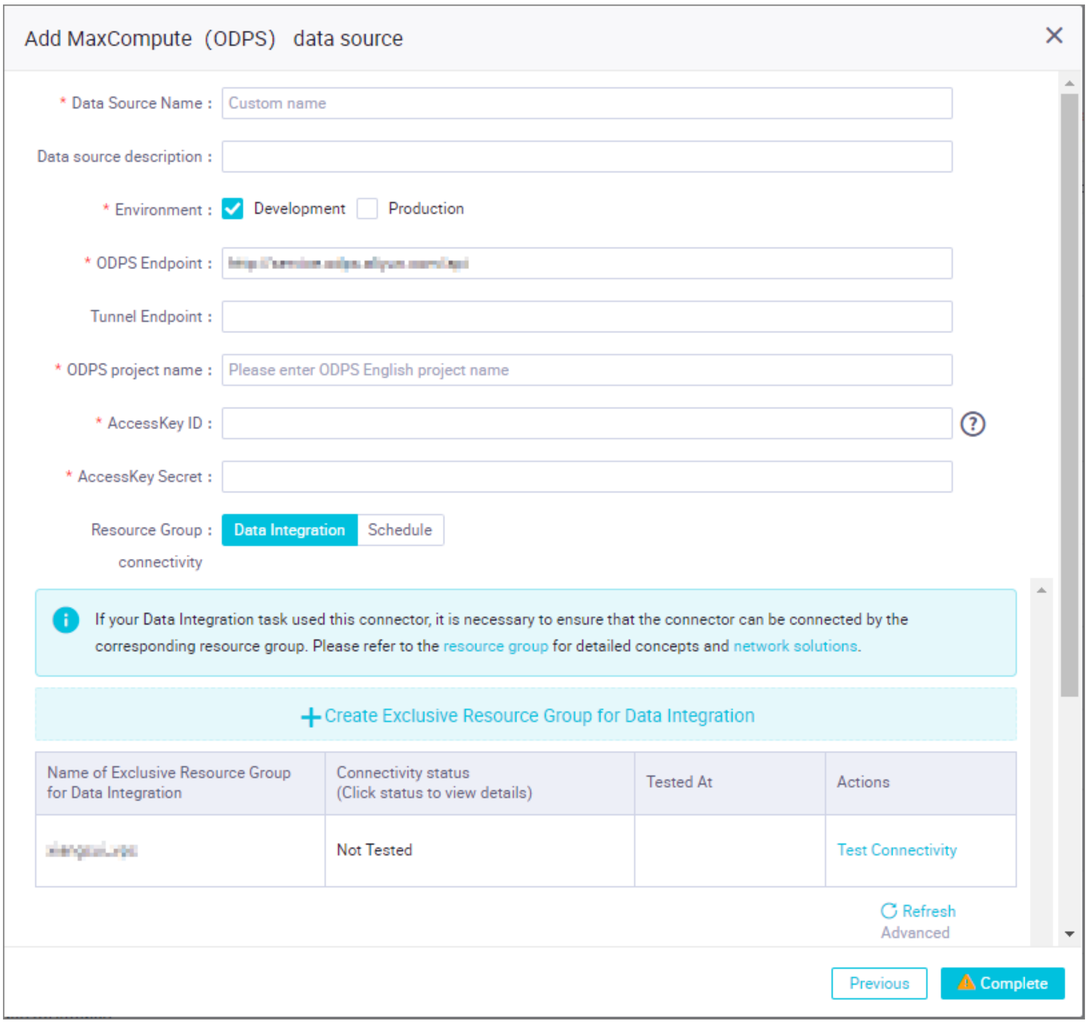
MaxCompute provides a comprehensive data import scheme that helps achieve fast computing of large amounts of data. If you associate a MaxCompute compute engine instance with a workspace for the first time, DataWorks generates the default data source odps_first for the workspace. Each time you associate a new MaxCompute compute engine instance with the workspace, DataWorks generates a compute engine data source named in the format of 0_Region ID_Compute engine instance name.

The MaxCompute project names of the default data source and the default compute engine data sources are the same as the names of the MaxCompute projects that are associated with the workspace. You can change the AccessKey pair of the default data source. To change the AccessKey pair, perform the following steps: In the DataWorks console, move the pointer over the profile picture in the upper-right corner and select AccessKey Management. On the AccessKey Management page, find the AccessKey ID that you want to enable or disable and click Enable or Disable in the Actions column. Take note of the following rules when you change the AccessKey pair:

- You can change only the AccessKey pair of one Alibaba Cloud account to that of another Alibaba Cloud account.
- Before you change the AccessKey pair, make sure that no Data Integration or DataStudio nodes are running in DataWorks. You can use a RAM user to access the MaxCompute data sources that you add.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **MaxCompute** in the Big Data Storage section.
4. In the **Add MaxCompute data source** dialog box, configure the parameters.



Add MaxCompute (ODPS) data source

- * Data Source Name : Custom name
- Data source description :
- * Environment : ☑ Development □ Production
- * ODPS Endpoint : [redacted]
- Tunnel Endpoint :
- * ODPS project name : Please enter ODPS English project name
- * AccessKey ID :
- * AccessKey Secret :
- Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+ Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
[redacted]	Not Tested		Test Connectivity

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
ODPS Endpoint	The endpoint of the MaxCompute project. The endpoint is automatically generated by MaxCompute based on system configurations.
Tunnel Endpoint	The endpoint of the MaxCompute Tunnel service. For more information, see Endpoints.
ODPS project name	The name of the MaxCompute project.
AccessKey ID	The AccessKey ID of the account that you use to connect to the MaxCompute project. You can view the AccessKey ID on the Security Management page.
AccessKey Secret	The AccessKey secret that corresponds to the AccessKey ID. The AccessKey secret is equivalent to a logon password.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > ⓘ Note
   >
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a MaxCompute data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure MaxCompute Reader and MaxCompute Writer. For more information, see MaxCompute Reader and MaxCompute Writer.

5.1.24. Add a Hive data source

DataWorks provides Hive Reader and Hive Writer for you to read data from and write data to Hive data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for Hive data sources.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information about the feature, see Isolate connections between the development and production environments.

If you use Object Storage Service (OSS) as the storage, you must take note of the following items:

- The value of the defaultFS parameter must start with oss://. For example, the value can be `oss://IP:PORT` or `oss://nameservice`.
- You must configure the parameters that are required for connecting to OSS in the advanced parameters of Hive. The following code provides an example:

```
{
            "hiveConfig":{
                "fs.oss.accessKeyId":"<yourAccessKeyId>",
                    "fs.oss.accessKeySecret":"<yourAccessKeySecret>",
                    "fs.oss.endpoint":"oss-cn-<yourRegion>-internal.aliyuncs.com"
            }
        }
```

Limits

- You can use only exclusive resource groups for Data Integration to read data from or write data to Hive data sources. For more information about exclusive resource groups for Data Integration, see Create and use an exclusive resource group for Data Integration.
- Hive data sources support only Kerberos authentication. Other authentication methods will be available in the future.

Add a Hive data source

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.

ii. In the left-side navigation pane, click **Workspaces**.

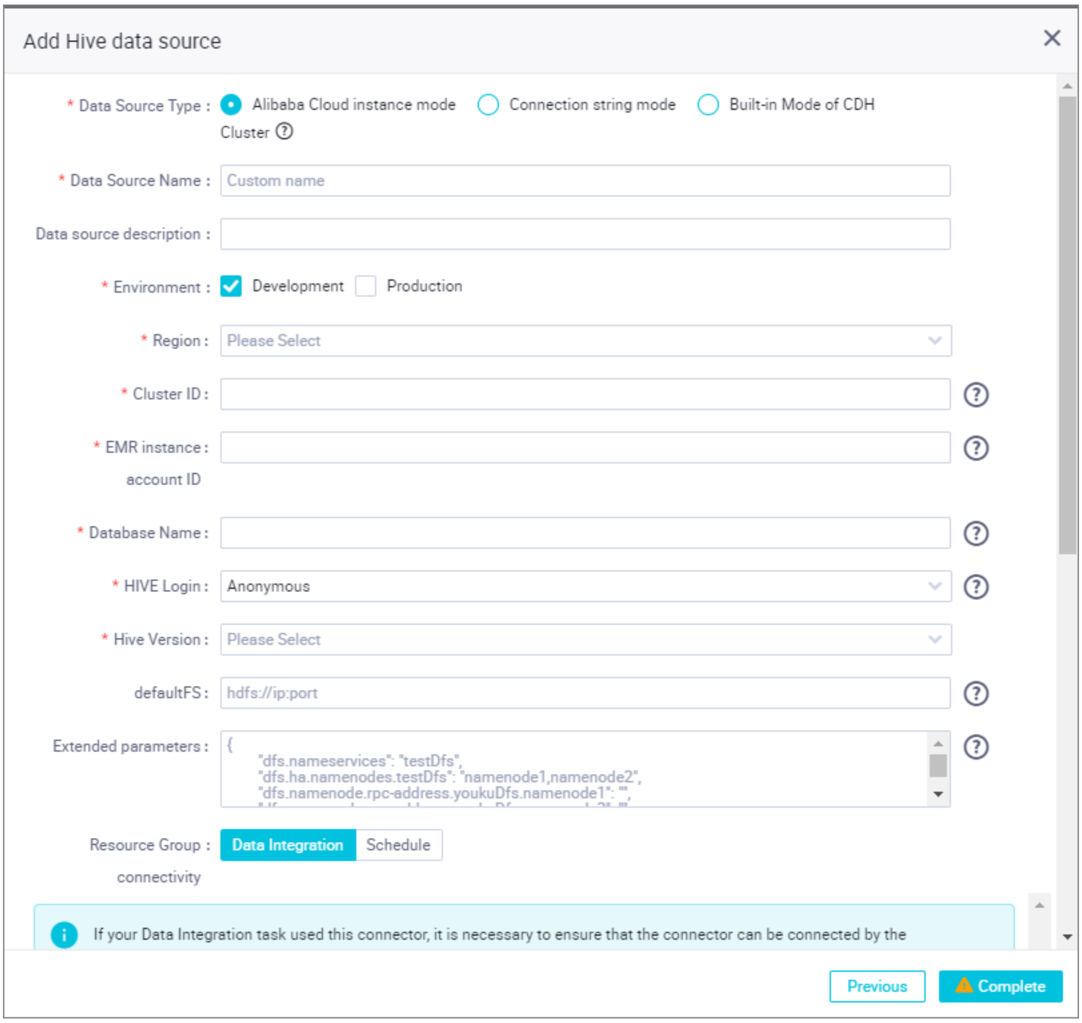
iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.

iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.

2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **Hive** in the Big Data Storage section.
4. In the **Add Hive data source** dialog box, configure the parameters.

   You can use one of the following modes to add a Hive data source: **Alibaba Cloud instance mode**, **Connection string mode**, and Built-in Mode of CDH.

   - The following table describes the parameters you must configure if you add a **Hive** data source by using **Alibaba Cloud instance mode**.



Add Hive data source

* Data Source Type : ● Alibaba Cloud instance mode ○ Connection string mode ○ Built-in Mode of CDH Cluster

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Region : Please Select

* Cluster ID :

* EMR instance account ID :

* Database Name :

* HIVE Login : Anonymous

* Hive Version : Please Select

defaultFS : hdfs://ip:port

Extended parameters :

```
{
        "dfs.nameservices": "testDfs",
        "dfs.ha.namenodes.testDfs": "namenode1,namenode2",
        "dfs.namenode.rpc-address.youkuDfs.namenode1": "",
```

Resource Group connectivity : Data Integration Schedule

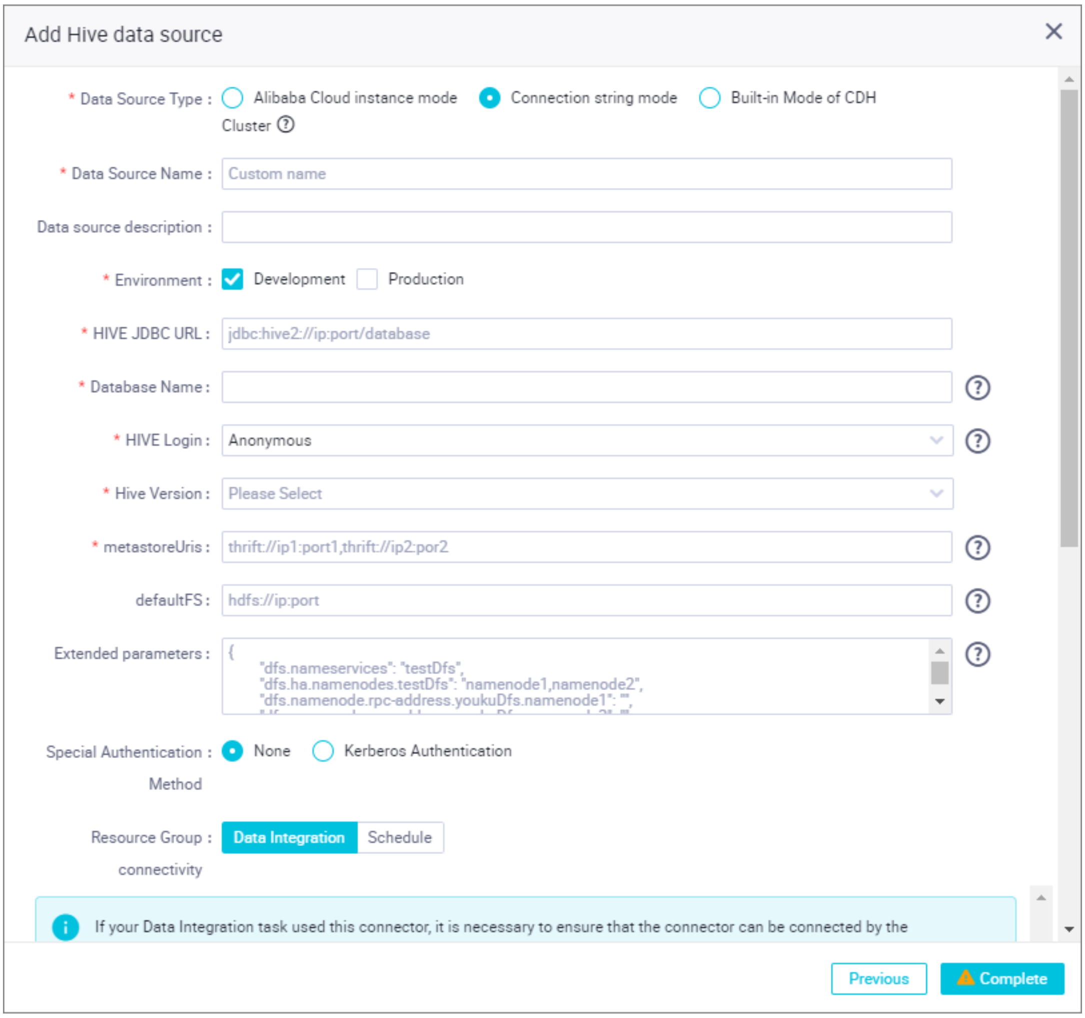
If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the

Parameter	Description
Data source type	The type of the data source. Set this parameter to **Alibaba Cloud instance mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.

Parameter	Description
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Region	The region where the data source resides.
Cluster ID	The ID of the E-MapReduce (EMR) cluster. You can log on to the EMR console to obtain the ID.
EMR instance account ID	The ID of the Alibaba Cloud account that is used to purchase the EMR cluster. You can log on to the Alibaba Cloud Management Console by using your Alibaba Cloud account and view your account ID on the **Security Settings** page.
Database Name	The name of the Hive database.
HIVE Login	The mode that is used to connect to the Hive database. Valid values: **Login with username and password** and **Anonymous**. If you select **Login with username and password**, enter the username and password that you can use to connect to the Hive database.
Hive Version	The Hive version that you want to use.
defaultFS	The address of the NameNode node in the Active state in Hadoop Distributed File System (HDFS), in the format of `hdfs://IP address of the host:Port number`.

Extended parameters	The advanced parameters of Hive, such as those related to high availability. The following code provides an example:
	`"hadoopConfig":{`
	`"dfs.nameservices": "testDfs",`
	`"dfs.ha.namenodes.testDfs": "namenode1,namenode2",`
	`"dfs.namenode.rpc-address.youkuDfs.namenode1": "",`
	`"dfs.namenode.rpc-address.youkuDfs.namenode2": "",`
	`"dfs.client.failover.proxy.provider.testDfs`
	`"org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyPro`
	`vider"`
	`}`

- The following table describes the parameters you must configure if you add a **Hive** data source by using **Connection string mode**.



Add Hive data source ✕

* Data Source Type : ○ Alibaba Cloud instance mode ● Connection string mode ○ Built-in Mode of CDH Cluster ⓘ

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* HIVE JDBC URL : jdbc:hive2://ip:port/database

* Database Name : ⓘ

* HIVE Login : Anonymous ⓘ

* Hive Version : Please Select

* metastoreUris : thrift://ip1:port1,thrift://ip2:por2 ⓘ

defaultFS : hdfs://ip:port ⓘ

Extended parameters : ⓘ

```
{
        "dfs.nameservices": "testDfs",
        "dfs.ha.namenodes.testDfs": "namenode1,namenode2",
        "dfs.namenode.rpc-address.youkuDfs.namenode1": "",
```

Special Authentication Method : ● None ○ Kerberos Authentication

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the

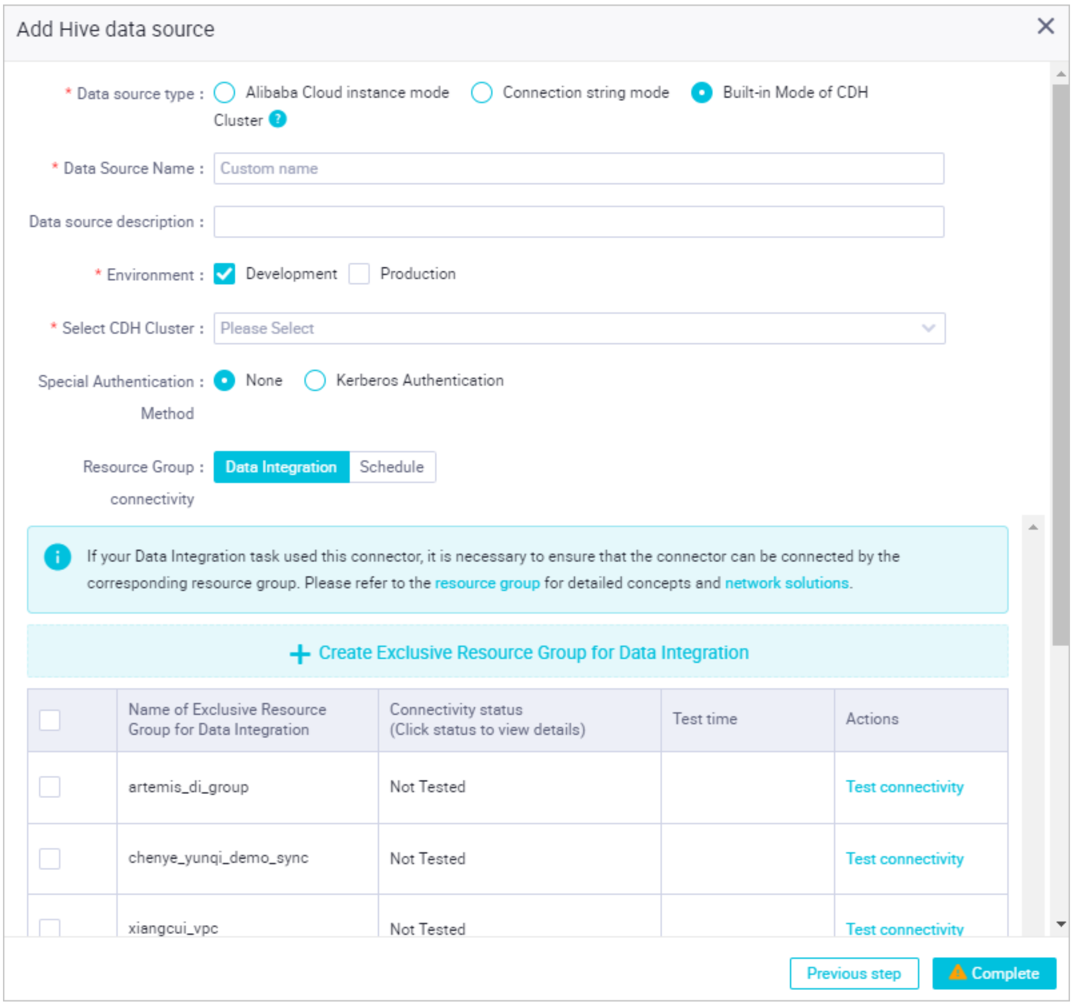
Parameter	Description
Data source type	The type of the data source. Set this parameter to **Connection string mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
HIVE JDBC URL	The Java Database Connectivity (JDBC) URL of the Hive metadatabase.
Database Name	The name of the Hive database. You can run the `show databases` command on the Hive client to query the created databases.

Parameter	Description
HIVE Login	The mode that is used to connect to the Hive database. Valid values: **Login with username and password** and **Anonymous**. If you select **Login with username and password**, enter the username and password that you can use to connect to the Hive database.
Hive Version	The Hive version that you want to use.
metastoreUris	The Uniform Resource Identifiers (URIs) of the Hive metadatabase, in the format of `thrift://ip1:port1,thrift://ip2:port2`.
defaultFS	The address of the NameNode node in the Active state in Hadoop Distributed File System (HDFS), in the format of `hdfs://IP address of the host:Port number`.

Extended parameters	The advanced parameters of Hive, such as those related to high availability. The following code provides an example:
	`"hadoopConfig":{`
	`"dfs.nameservices": "testDfs",`
	`"dfs.ha.namenodes.testDfs": "namenode1,namenode2",`
	`"dfs.namenode.rpc-address.youkuDfs.namenode1": "",`
	`"dfs.namenode.rpc-address.youkuDfs.namenode2": "",`
	`"dfs.client.failover.proxy.provider.testDfs`
	`"org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider"`
	`}`

Special Authentication Method	Specifies whether to enable identity authentication. Default value: **None**. You can alternatively set this parameter to **Kerberos Authentication**. For more information about Kerberos authentication, see Configure Kerberos authentication.
Keytab File	If you set the **Special Authentication Method** parameter to **Kerberos Authentication**, you must select the specified keytab from the Keytab File drop-down list. If no keytab is available, you can click **Add Authentication File** to upload a keytab.
CONF File	If you set the **Special Authentication Method** parameter to **Kerberos Authentication**, you must select the specified CONF file from the CONF File drop-down list. If no CONF file is available, you can click **Add Authentication File** to upload a CONF file.
principal	The Kerberos principal. Specify this parameter in the format of Principal name/Instance name@Domain name, such as ****/hadoopclient@**.***.

- The following table describes the parameters you must configure if you add a **Hive** data source by using **Built-in Mode of CDH Cluster**.



Add Hive data source ✕

* Data source type : ○ Alibaba Cloud instance mode ○ Connection string mode ● Built-in Mode of CDH Cluster

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Select CDH Cluster : Please Select

Special Authentication Method : ● None ○ Kerberos Authentication

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+ Create Exclusive Resource Group for Data Integration

□	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Test time	Actions
□	artemis_di_group	Not Tested		Test connectivity
□	chenye_yunqi_demo_sync	Not Tested		Test connectivity
□	xiangcui_vpc	Not Tested		Test connectivity

Parameter	Description
Data source type	The type of the data source. Set this parameter to **Built-in Mode of CDH**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Select CDH Cluster	The CDH cluster that you want to use.
Special Authentication Method	Specifies whether to enable identity authentication. Default value: **None**. You can alternatively set this parameter to **Kerberos Authentication**. For more information about Kerberos authentication, see Configure Kerberos authentication.

Parameter	Description
Keytab File	If you set the **Special Authentication Method** parameter to **Kerberos Authentication**, you must select the specified keytab from the Keytab File drop-down list. If no keytab is available, you can click **Add Authentication File** to upload a keytab.
CONF File	If you set the **Special Authentication Method** parameter to **Kerberos Authentication**, you must select the specified CONF file from the CONF File drop-down list. If no CONF file is available, you can click **Add Authentication File** to upload a CONF file.
principal	The Kerberos principal. Specify this parameter in the format of Principal name/Instance name@Domain name, such as ****/hadoopclient@**.***.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

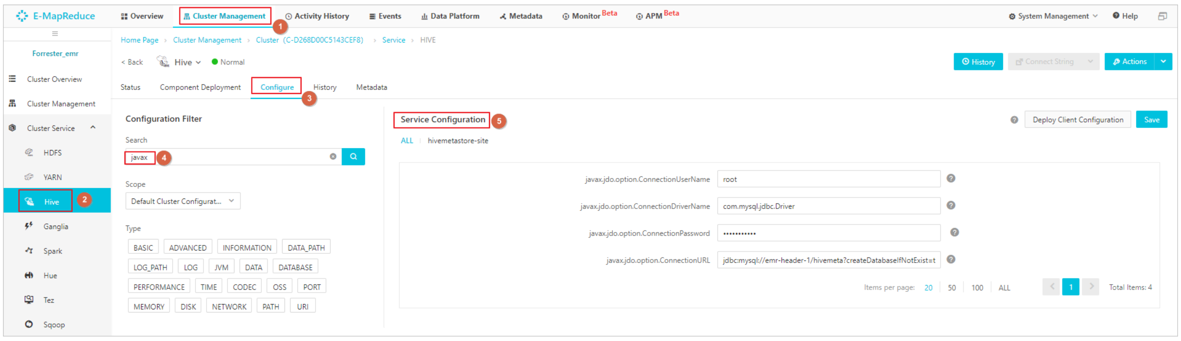
   Note

   - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

Obtain the Hive configuration in the EMR console

1. Log on to the EMR console.
2. In the top navigation bar, click **Cluster Management**.
3. On the Cluster Management tab, find the cluster whose details you want to view and click **Details** in the Actions column. On the **Cluster Overview** page, view the cluster details.
4. In the left-side navigation pane, choose **Cluster Service > Hive**.
5. On the Hive page, click the **Configure** tab.
6. In the **Configure Filter** section, enter **javax** in the search box and click the 🔍 icon to view the Hive configuration in the **Service Configuration** section.



Search: javax

javax.jdo.option.ConnectionUserName: root

javax.jdo.option.ConnectionDriverName: com.mysql.jdbc.Driver

javax.jdo.option.ConnectionPassword: ••••••••••••

javax.jdo.option.ConnectionURL: jdbc:mysql://emr-header-1/hivemeta?createDatabaseIfNotExist=t

5.1.25. Add a GBase8a data source

DataWorks provides GBase8a Reader and GBase8a Writer for you to read data from and write data to GBase8a data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for GBase8a data sources.

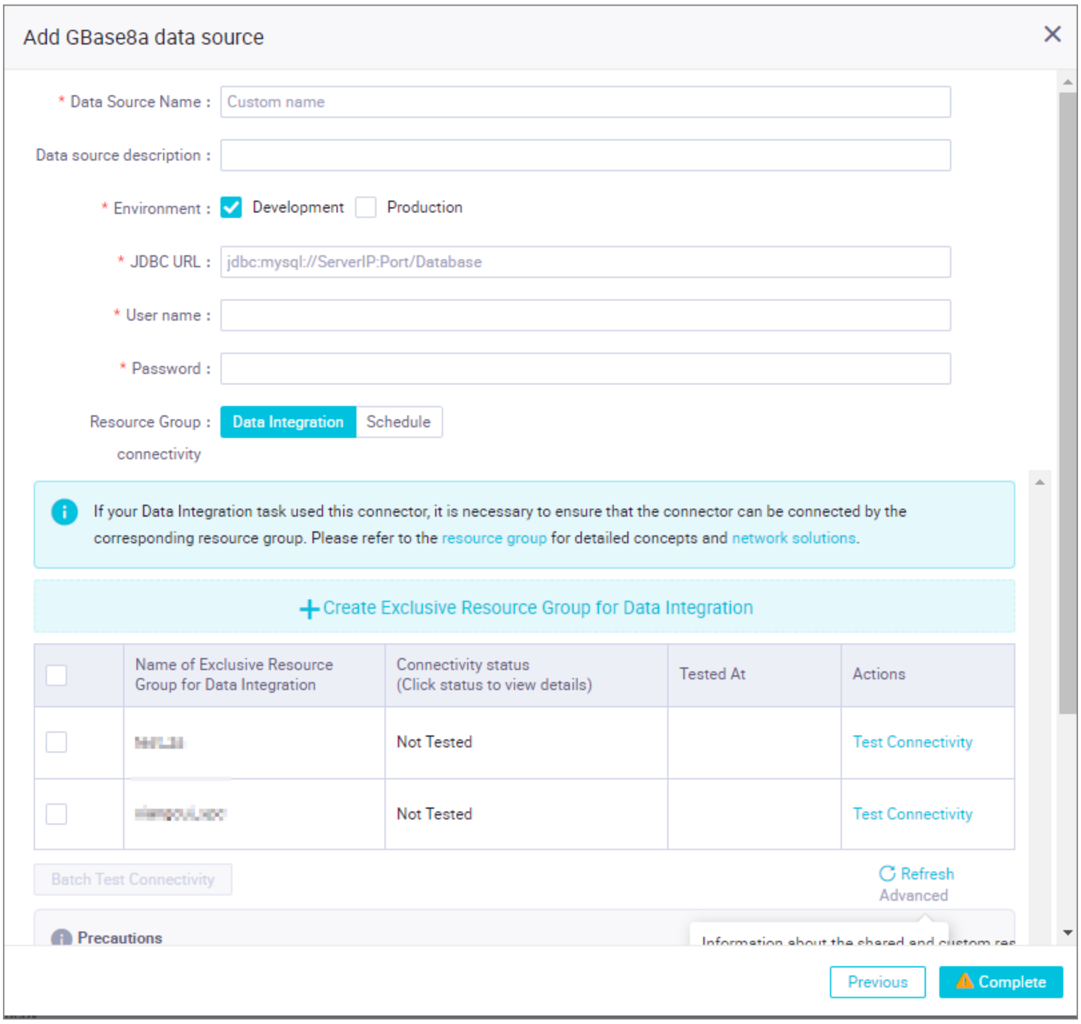
Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

> **Notice** GBase8a data sources support only exclusive resource groups for Data Integration and custom resource groups for Data Integration, but not shared resource groups. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **GBase8a** in the Big Data Storage section.
4. In the **Add GBase8a data source** dialog box, configure the parameters.



Add GBase8a data source

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* JDBC URL : jdbc:mysql://ServerIP:Port/Database

* User name :

* Password :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+ Create Exclusive Resource Group for Data Integration

□	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
□	[redacted]	Not Tested		Test Connectivity
□	[redacted]	Not Tested		Test Connectivity

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database. Specify this parameter in the format of `jdbc:mysql://Server IP address:Port number/Database` .
User name	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be

normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

> Note
>
> - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
> - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a GBase8a data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure GBase8a Reader and GBase8a Writer. For more information, see Gbase8a Reader and GBase8a Writer.

5.1.26. Add a Hologres data source

DataWorks provides Hologres Reader and Hologres Writer for you to read data from and write data to Hologres data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for Hologres data sources.

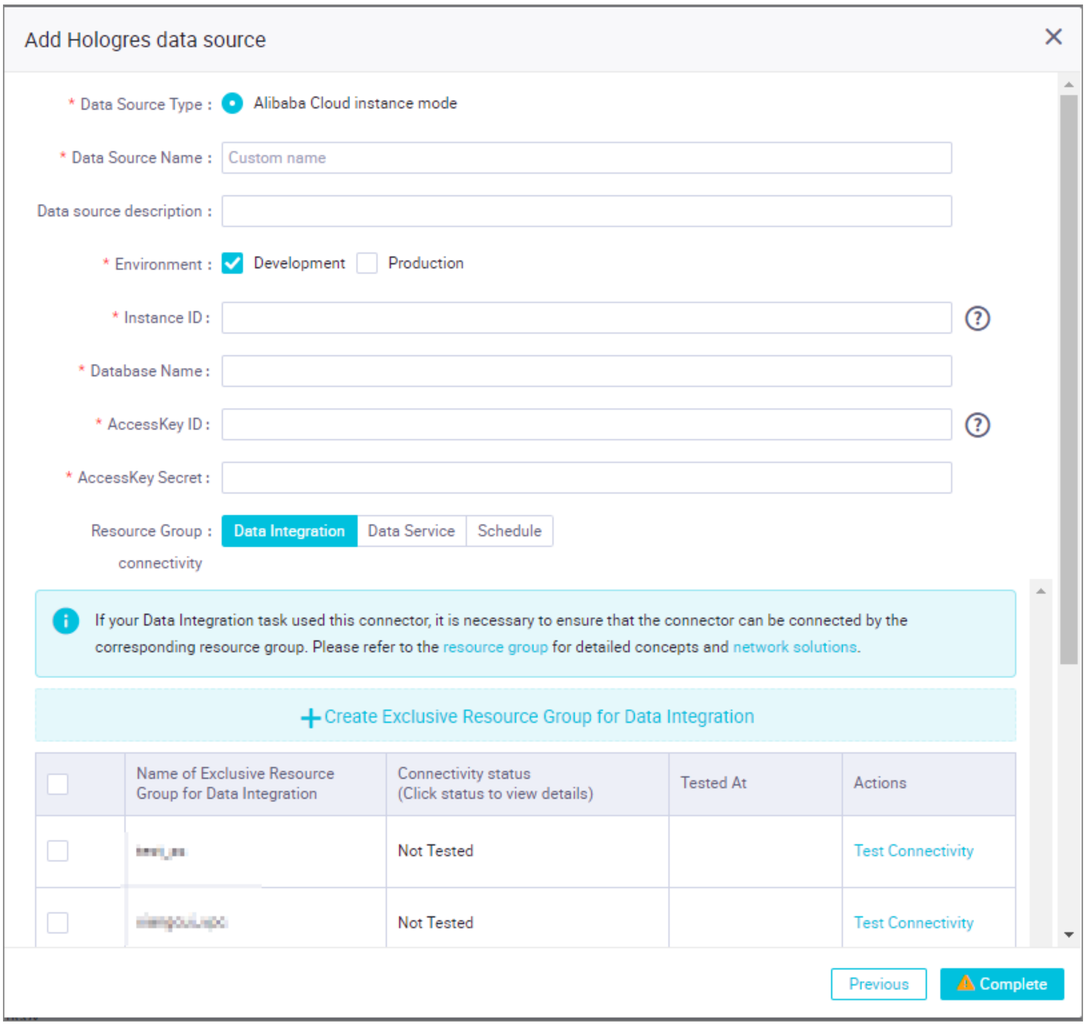
Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

> Notice Hologres data sources support only exclusive resource groups for Data Integration, but not the default resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **Hologres** in the Big Data Storage section.
4. In the **Add Hologres data source** dialog box, configure the parameters.



Add Hologres data source

* Data Source Type : ● Alibaba Cloud instance mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Instance ID :

* Database Name :

* AccessKey ID :

* AccessKey Secret :

Resource Group connectivity : Data Integration | Data Service | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

□	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
□	[redacted]	Not Tested		Test Connectivity
□	[redacted]	Not Tested		Test Connectivity

Parameter	Description
Data source type	The type of the data source. The value of this parameter can be only **Alibaba Cloud instance mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Instance ID	The ID of the Hologres instance. You can log on to the Hologres console to view the ID.
Database Name	The name of the Hologres database.
AccessKey ID	The AccessKey ID of the account that you can use to connect to the Hologres database. You can view the AccessKey ID on the Security Management page.
AccessKey Secret	The AccessKey secret of the account that you can use to connect to the Hologres database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > ⓘ Note
   >
   > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a Hologres data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure Hologres Reader and Hologres Writer. For more information, see Hologres Reader or Hologres Writer.

5.1.27. Add an HBase data source

DataWorks provides HBase Reader and HBase Writer for you to read data from and write data to HBase data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for HBase data sources.

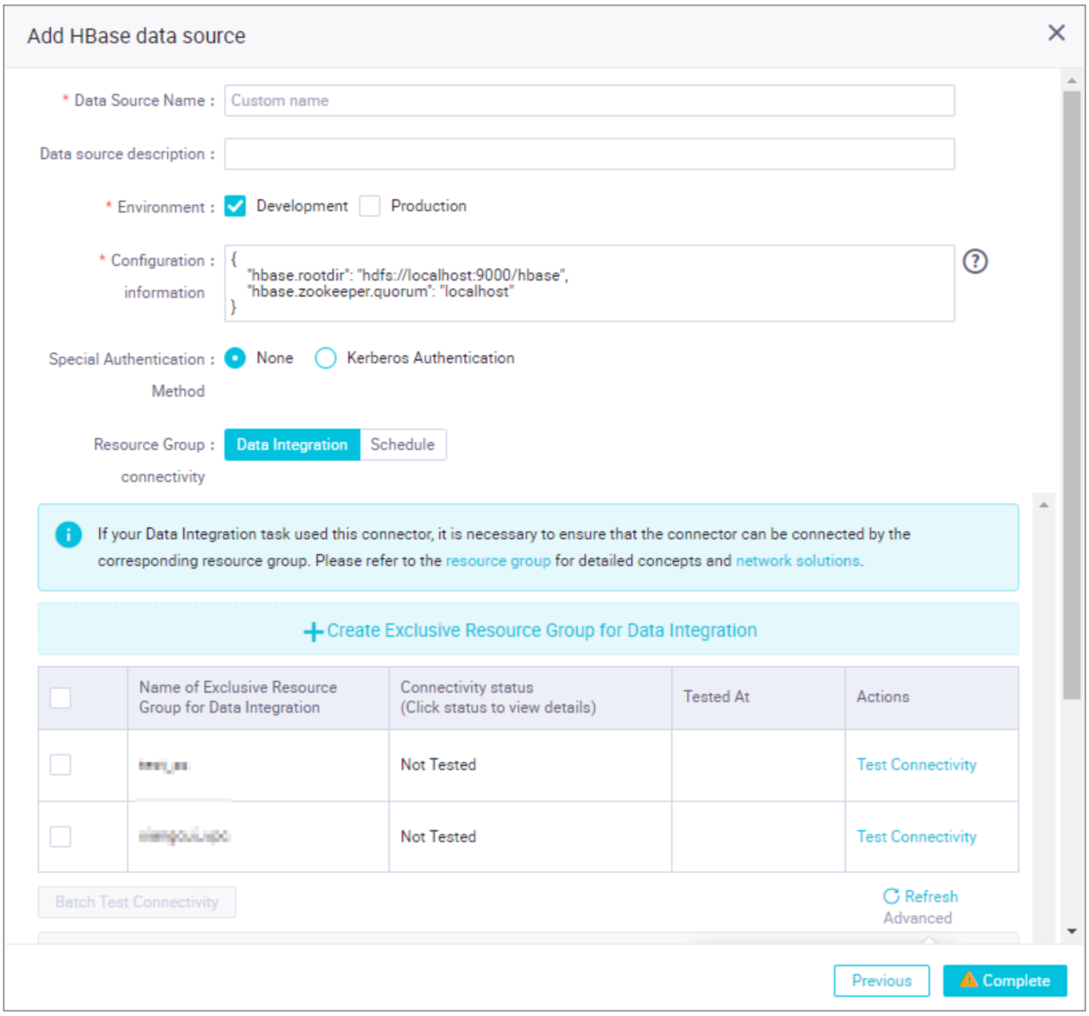
Limits

- HBase data sources support only Kerberos authentication. Other authentication methods will be available in the future.
- The following table describes the network connection methods supported by HBase of different editions and versions.

Edition and version	Internet connection	VPC connection
Standard Edition (1.1 and 2.0)	Not supported	Supported
Performance-enhanced Edition	Supported	Supported

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **HBase** in the Big Data Storage section.
4. In the **Add HBase data source** dialog box, configure the parameters.



Add HBase data source

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Configuration information :

```
{
  "hbase.rootdir": "hdfs://localhost:9000/hbase",
  "hbase.zookeeper.quorum": "localhost"
}
```

Special Authentication Method : ● None ○ Kerberos Authentication

Resource Group connectivity : Data Integration Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

□	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
□	[redacted]	Not Tested		Test Connectivity
□	[redacted]	Not Tested		Test Connectivity

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data Source Description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.

Parameter	Description

Configuration Information	The configuration information of the HBase cluster.
	You can convert the hbase-site.xml file to the JSON format. Then, add HBase client properties, such as cache and batch for scan, to optimize the interaction between the cluster and the client.
	You must configure different information based on the edition of HBase in use.
	Note Standard Edition (1.1 and 2.0) and Performance-enhanced Edition are supported. For more information about the editions, see ApsaraDB for HBase editions.
	○ If you are using the Standard Edition, the default configuration is used. You need only to enter the related ZooKeeper information.
	`{`
	`    "hbase.rootdir": "hdfs://localhost:9000/hbase",`
	`    "hbase.zookeeper.quorum": "localhost"`
	`}`
	○ If you are using the Performance-enhanced Edition, the endpoint parameter specific to this edition is required, and the zookeeper.quorum parameter is not required.
	The following code provides an example on how to add an HBase data source of the Performance-enhanced Edition (Lindorm):
	`Enter the following configurations in the Configuration information field:`
	`{`
	`"hbase.client.connection.impl" :`
	`"com.alibaba.hbase.client.AliHBaseUEConnection",`
	`"hbase.client.endpoint" : "host:30020",`
	`"hbase.client.username" : "root",`
	`"hbase.client.password" : "root"`
	`}`

Special Authentication Method	Specifies whether to enable identity authentication. Default value: **None**. You can alternatively set this parameter to **Kerberos Authentication**. For more information about Kerberos authentication, see Configure Kerberos authentication.
Keytab File	If you set the **Special Authentication Method** parameter to **Kerberos Authentication**, you must select the specified keytab from the Keytab File drop-down list. If no keytab is available, you can click **Add Authentication File** to upload a keytab.
CONF File	If you set the **Special Authentication Method** parameter to **Kerberos Authentication**, you must select the specified CONF file from the CONF File drop-down list. If no CONF file is available, you can click **Add Authentication File** to upload a CONF file.
principal	The Kerberos principal. Specify this parameter in the format of Principal name/Instance name@Domain name, such as ****/hadoopclient@**.***.

5. Set **Resource Group connectivity** to **Data Integration**.

6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the network connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. For more information, see Select a network connectivity solution.

   > Note
   >
   > - Connectivity tests can be performed only for exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.
   > - If you are using the Performance-enhanced Edition (Lindorm), an error message indicating that the AliHBase class cannot be found appears when you test the network connectivity of your resource group. You can ignore this error message.

7. After the data source passes the connectivity test, click **Complete**.

> Note The test for network connectivity between an HBase data source and a resource group may fail. To resolve this issue, DataWorks allows you to specify the version of the HBase data source that you use. This indicates that you must configure the hbaseVersion parameter when you add an HBase data source. The following code provides an example:
>
> ```
> {
> "hbase.zookeeper.quorum": "my-zk:2181",
> "hbaseVersion": "2.0.14"
> }
> ```

What's next

You have learned how to add an HBase data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure HBase Reader and HBase Writer. For more information, see HBase Reader and HBase Writer.

5.1.28. Add an Elasticsearch data source

DataWorks provides Elasticsearch Reader and Elasticsearch Writer for you to read data from and write data to Elasticsearch data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for Elasticsearch data sources.

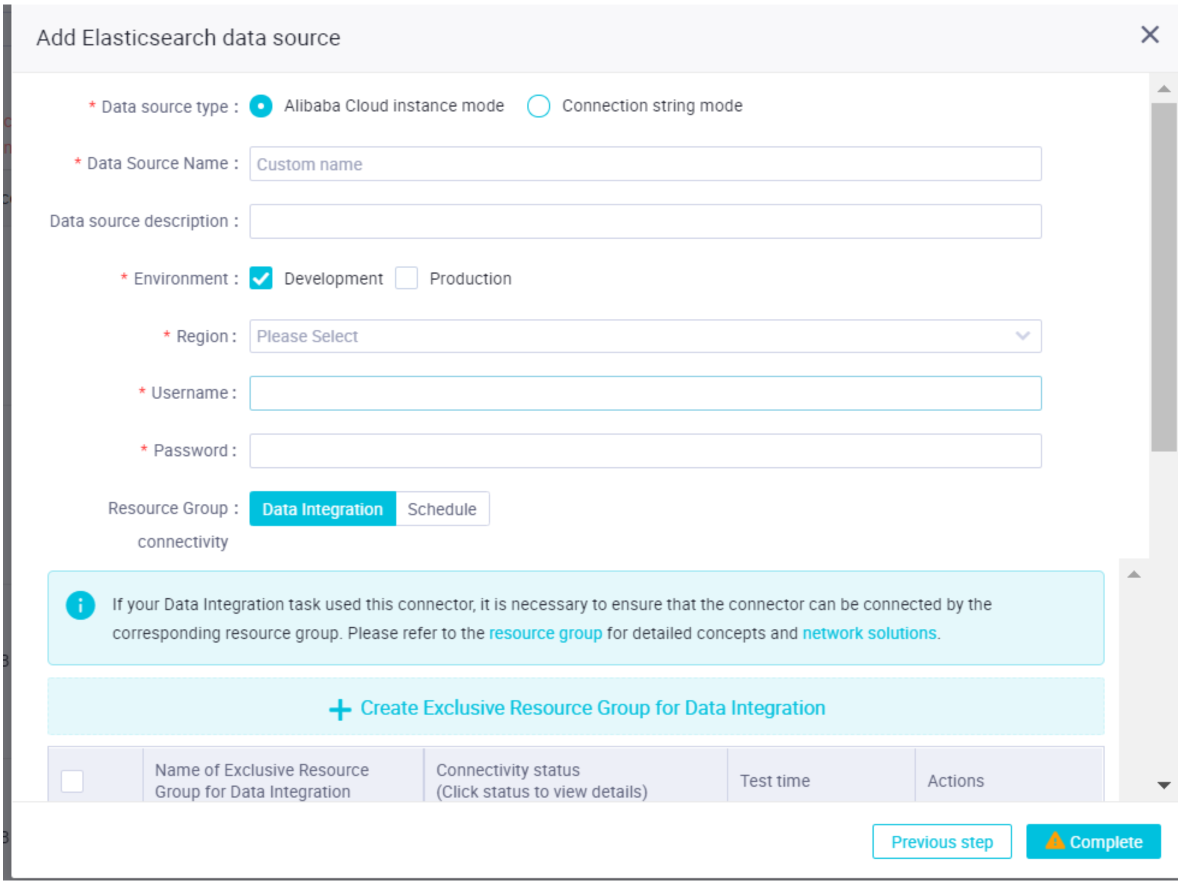
Context

DataWorks allows you to add Alibaba Cloud Elasticsearch V5.X, V6.X, and V7.X clusters as data sources. Self-managed Elasticsearch clusters are not supported.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **Elasticsearch** in the NoSQL section.
4. In the **Add Elasticsearch data source** dialog box, configure the parameters.

   You can use one of the following modes to add an Elasticsearch data source: **Alibaba Cloud instance mode** and **Connection string mode**.



Add Elasticsearch data source

* Data source type : ● Alibaba Cloud instance mode ○ Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Region : Please Select

* Username :

* Password :

Resource Group connectivity : Data Integration Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

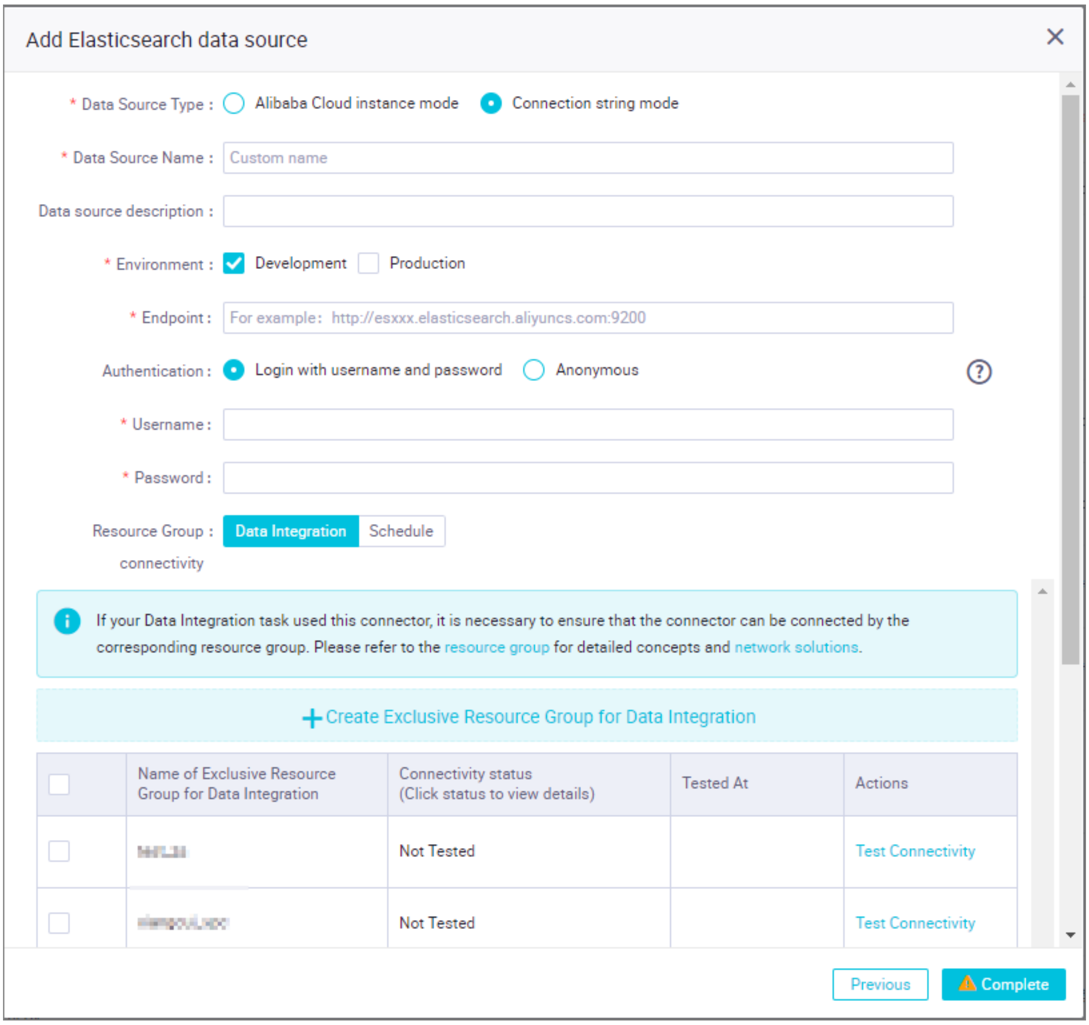
+ Create Exclusive Resource Group for Data Integration

□	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Test time	Actions

- The following table describes the parameter configurations of an Elasticsearch data source added in **Alibaba Cloud instance mode**.

Parameter	Description
Data source type	The mode in which the data source is added. Set this parameter to **Alibaba Cloud instance mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. Note This parameter is displayed only when the workspace is in standard mode.
Region	The region where the data source resides.
Username	The username that is used to connect to the data source.
Password	The password that is used to connect to the data source.

- The following table describes the parameter configurations of an Elasticsearch data source added in **Connection string mode**.



Add Elasticsearch data source ✕

* Data Source Type : ○ Alibaba Cloud instance mode ● Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Endpoint : For example: http://esxxx.elasticsearch.aliyuncs.com:9200

Authentication : ● Login with username and password ○ Anonymous

* Username :

* Password :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

□	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
□	[redacted]	Not Tested		Test Connectivity
□	[redacted]	Not Tested		Test Connectivity

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Endpoint	The endpoint of Elasticsearch. Specify a value for this parameter in the format of `http://esxxxx.elasticsearch.aliyuncs.com:9200`.
Authentication	The authentication mode. Valid values: **Login with username and password** and **Anonymous**.
Username	The username that is used to connect to the data source.
Password	The password that is used to connect to the data source.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test**

connectivity in the Actions column.

A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

> Note
>
> - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
> - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add an Elasticsearch data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure Elasticsearch Reader and Elasticsearch Writer. For more information, see Elasticsearch Reader and Elasticsearch Writer.

5.1.29. Add a Vertica data source

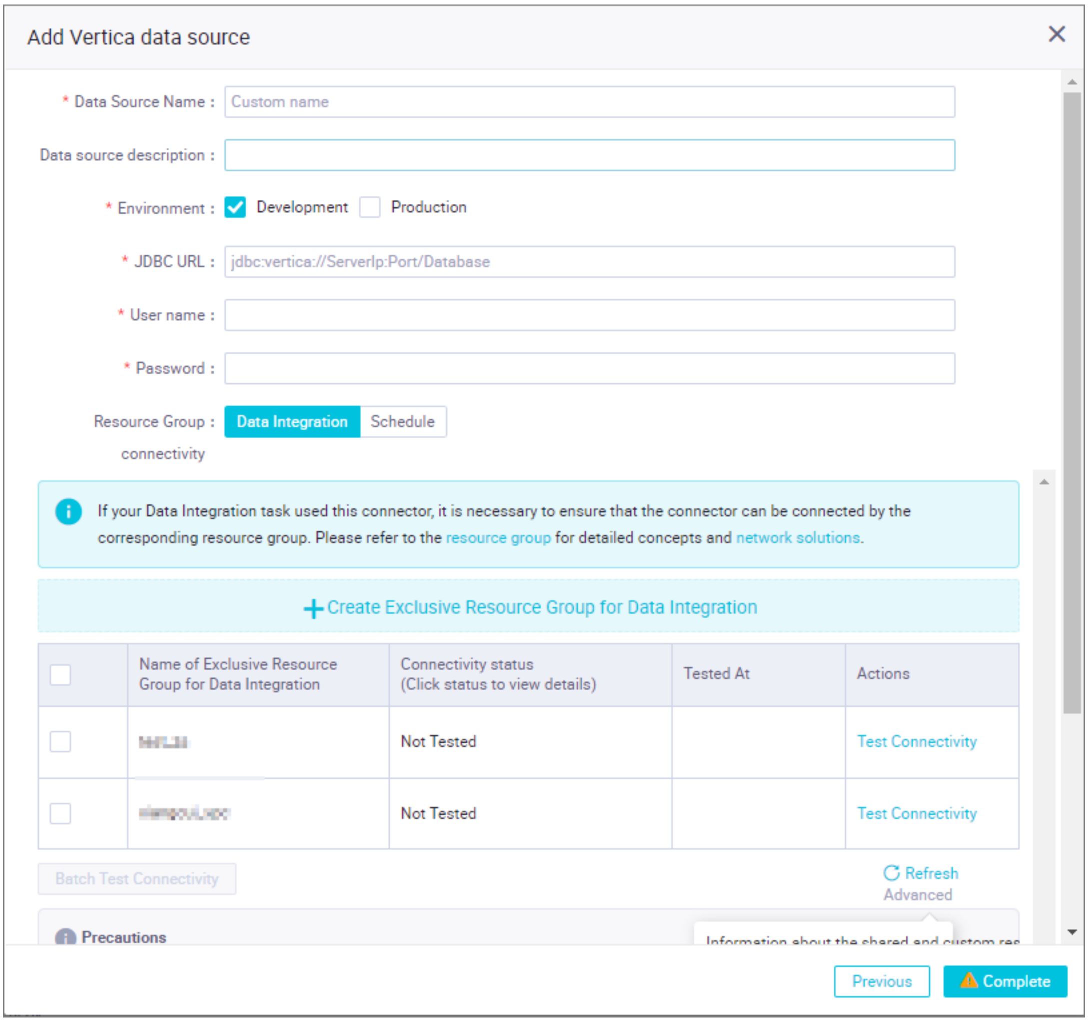
DataWorks provides Vertica Reader and Vertica Writer for you to read data from and write data to Vertica data sources. You can use the code editor to configure synchronization nodes for Vertica data sources.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **Vertica** in the Big Data Storage section.
4. In the **Add Vertica data source** dialog box, configure the parameters.



Add Vertica data source

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* JDBC URL : jdbc:vertica://ServerIp:Port/Database

* User name :

* Password :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+ Create Exclusive Resource Group for Data Integration

□	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
□	[redacted]	Not Tested		Test Connectivity
□	[redacted]	Not Tested		Test Connectivity

Batch Test Connectivity

Refresh
Advanced

Precautions

Information about the shared and custom res

Previous | Complete

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database. Specify this parameter in the format of `jdbc:vertica://Server IP address:Port number/Database`.
User name	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

5. Set **Resource Group connectivity** to **Data Integration**.

6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be

normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

> ⓘ Note
>
> - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
> - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add a Vertica data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure Vertica Reader and Vertica Writer. For more information, see Vertica Reader and Vertica Writer.

5.1.30. Add a RestAPI data source

If a data source can be connected by using APIs, you can use the RestAPI (HTTP) feature of Alibaba Cloud DataWorks to connect to the data source and integrate data. Before you use this feature to integrate data, you must add a RestAPI data source. This topic describes how to add a RestAPI data source.

Limits

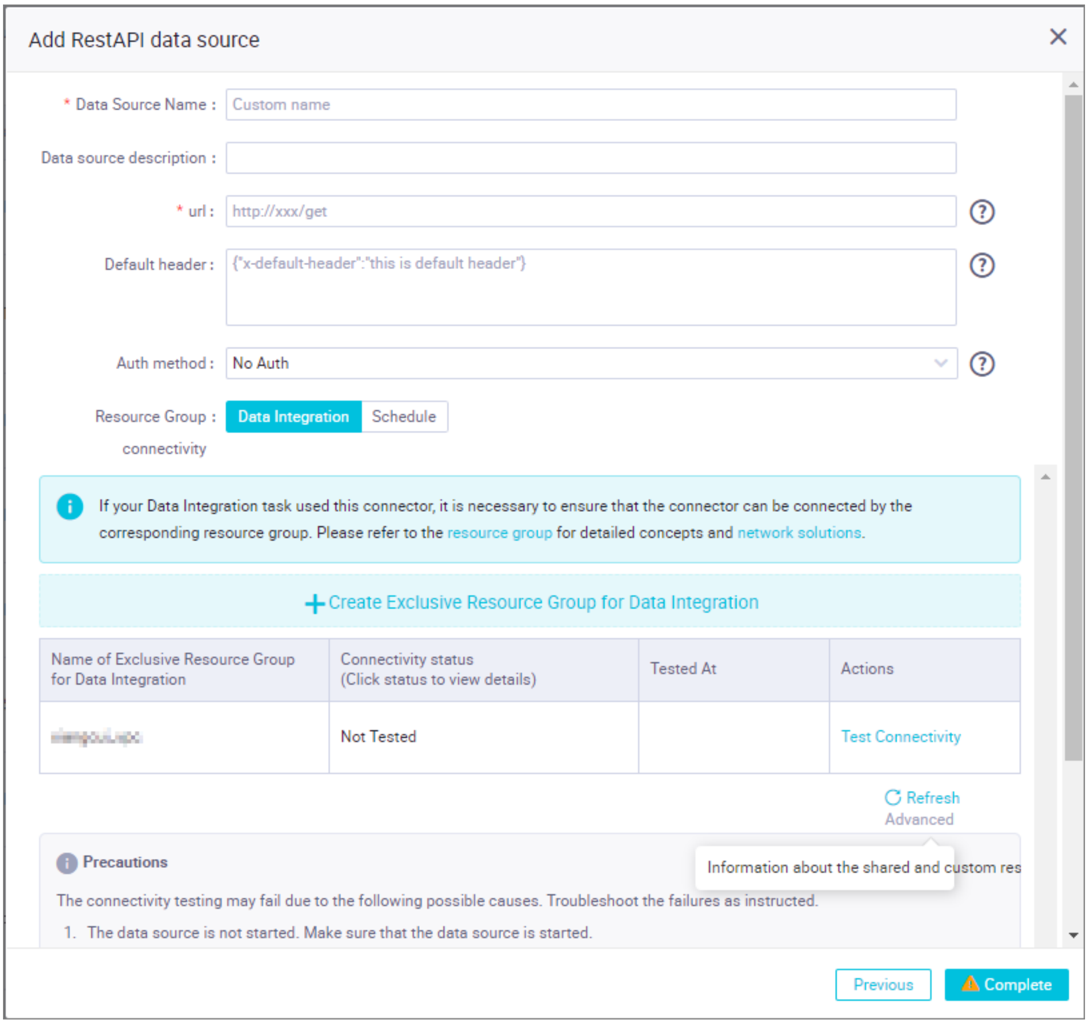
RestAPI data sources support only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.

Procedure

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

If you use a DataWorks workspace in standard mode, perform the following steps to add data sources separately for the development and production environments:

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **RestAPI** in the Semi-structured storage section.
4. In the Add RestAPI data source dialog box, configure the parameters.



Add RestAPI data source

* Data Source Name :	Custom name
Data source description :	
* url :	http://xxx/get
Default header :	{"x-default-header":"this is default header"}
Auth method :	No Auth
Resource Group connectivity :	Data Integration · Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
[redacted]	Not Tested		Test Connectivity

Refresh

Advanced

Information about the shared and custom res

Precautions

The connectivity testing may fail due to the following possible causes. Troubleshoot the failures as instructed.

1. The data source is not started. Make sure that the data source is started.

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Connection Type	The network connection type. Select a network connection type based on the network environment of the data source.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Url	The URL of the RESTful API.
Default header	The content of the header in each request that is transferred to the URL.

Parameter	Description
Auth method	The authentication method. The RestAPI feature supports the following authentication methods: Basic Auth, Token Auth, No Auth, and Aliyun API Signature. You must select an authentication method and configure the displayed authentication parameters based on the authentication method that is supported by the data source. ○ Basic Auth: basic authentication If the data source supports username- and password-based authentication, you can select Basic Auth and configure the username and password that can be used for authentication. During data integration, the username and password are transferred to the RESTful API URL for authentication. The data source is connected only after the authentication is successful. ○ Token Auth: token-based authentication If the data source supports token-based authentication, you can select Token Auth and configure a fixed token value that can be used for authentication. During data integration, the token is contained in the request header, such as {"Authorization":"Bearer TokenXXXXXX"}, and transferred to the RESTful API URL for authentication. The data source is connected only after the authentication is successful. ○ Aliyun API Signature: Alibaba Cloud API signature-based authentication If the following conditions are met, you can select Aliyun API Signature and configure the AccessKey ID and AccessKey secret that can be used for authentication: The data source that you want to connect is an Alibaba Cloud service, and the API of this service supports AccessKey pair-based authentication.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > **Note** You can use only exclusive resource groups. For more information about exclusive resource groups for Data Integration, see Create and use an exclusive resource group for Data Integration.

7. After the data source passes the connectivity test, click **Complete**.

What's next

After you add a RestAPI data source, you can configure a synchronization node for the data source by using the codeless user interface (UI) or code editor.

5.1.31. Add a SAP HANA data source

SAP HANA is an in-memory computing platform that can be deployed on premises or on the cloud. It features high-performance data queries. DataWorks provides SAP HANA Reader and SAP HANA Writer for you to read data from and write data to SAP HANA data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for SAP HANA data sources.

Limits

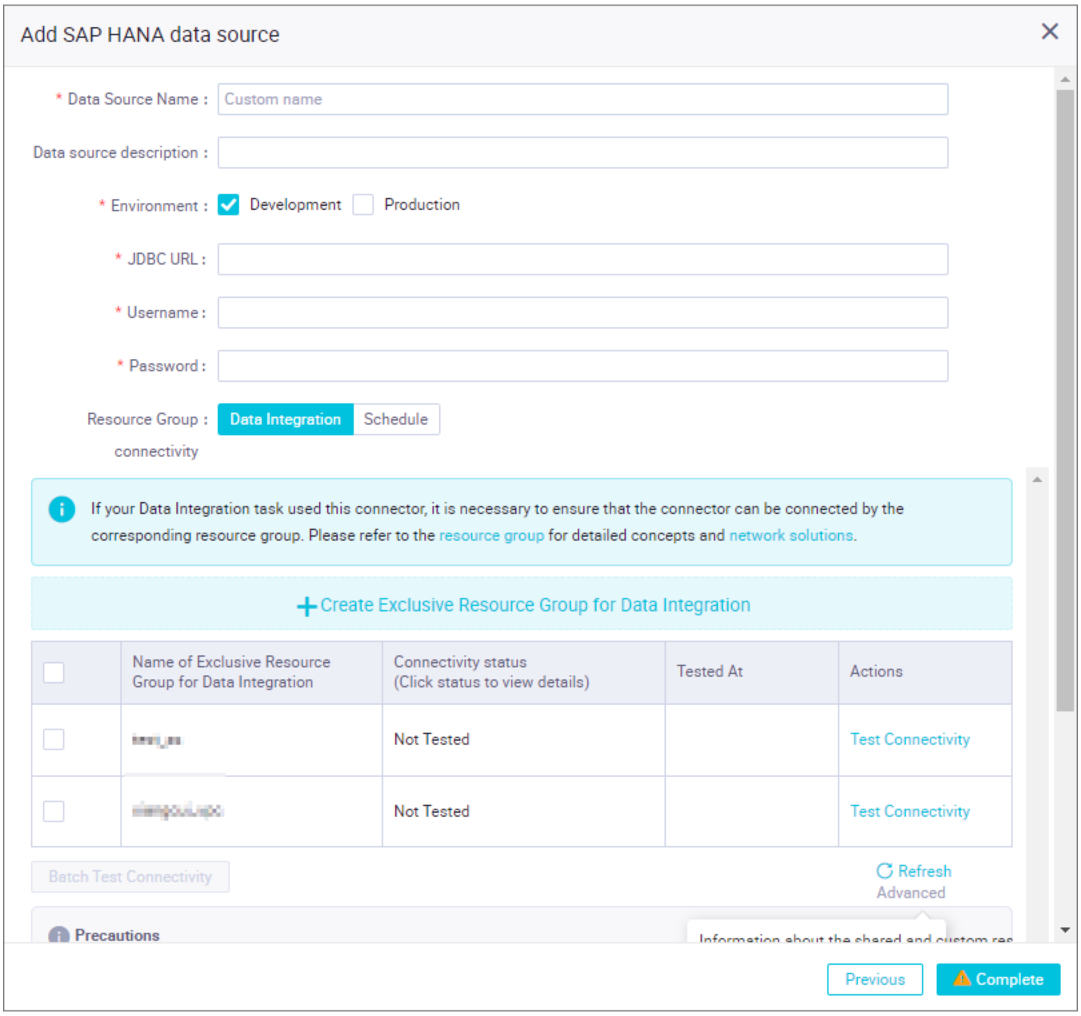
RestAPI data sources support only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.

Procedure

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

If you use a DataWorks workspace in standard mode, perform the following steps to add data sources separately for the development and production environments:

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **SAP HANA** in the Relational Database section.
4. In the **Add SAP HANA data source** dialog box, configure the parameters.



Add SAP HANA data source

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* JDBC URL :

* Username :

* Password :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

□	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions
□	[redacted]	Not Tested		Test Connectivity
□	[redacted]	Not Tested		Test Connectivity

Batch Test Connectivity

Refresh
Advanced

Precautions

Information about the shared and custom res

Previous Complete

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.

Parameter	Description
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. ⓘ **Note** This parameter is displayed only when the workspace is in standard mode.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database. Specify this parameter in the format of `jdbc:sap://host:Port?currentschema=SCHEMA` .
Username	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   ⓘ **Note** You can use only exclusive resource groups. For more information about exclusive resource groups for Data Integration, see Create and use an exclusive resource group for Data Integration.
7. After the data source passes the connectivity test, click **Complete**.

What's next

After you add a RestAPI data source, you can configure a synchronization node for the data source by using the codeless user interface (UI) or code editor.

5.1.32. Add a KingbaseES data source

DataWorks provides KingbaseES Reader and KingbaseES Writer for you to read data from and write data to KingbaseES data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for KingbaseES data sources. This topic describes how to add a KingbaseES data source in the DataWorks console to support subsequent data synchronization.

Context

KingbaseES is a large-scale relational database management system (DBMS) developed by Beijing Renda Jincang Information Technology Co., Ltd. under the support from major database projects in 863 Program and the support of Beijing Municipal Science and Technology Project. KingbaseES has independent intellectual property rights. KingbaseES provides complete database management features and supports more than 1,000 concurrent users, terabytes of data, and gigabytes of large-size objects. KingbaseES can run on various operating systems such as Windows, Linux, Kylin, and UNIX. KingbaseES has the following advantages: standard and general-purpose, stable and efficient, secure and reliable, and easy to use.

Limits

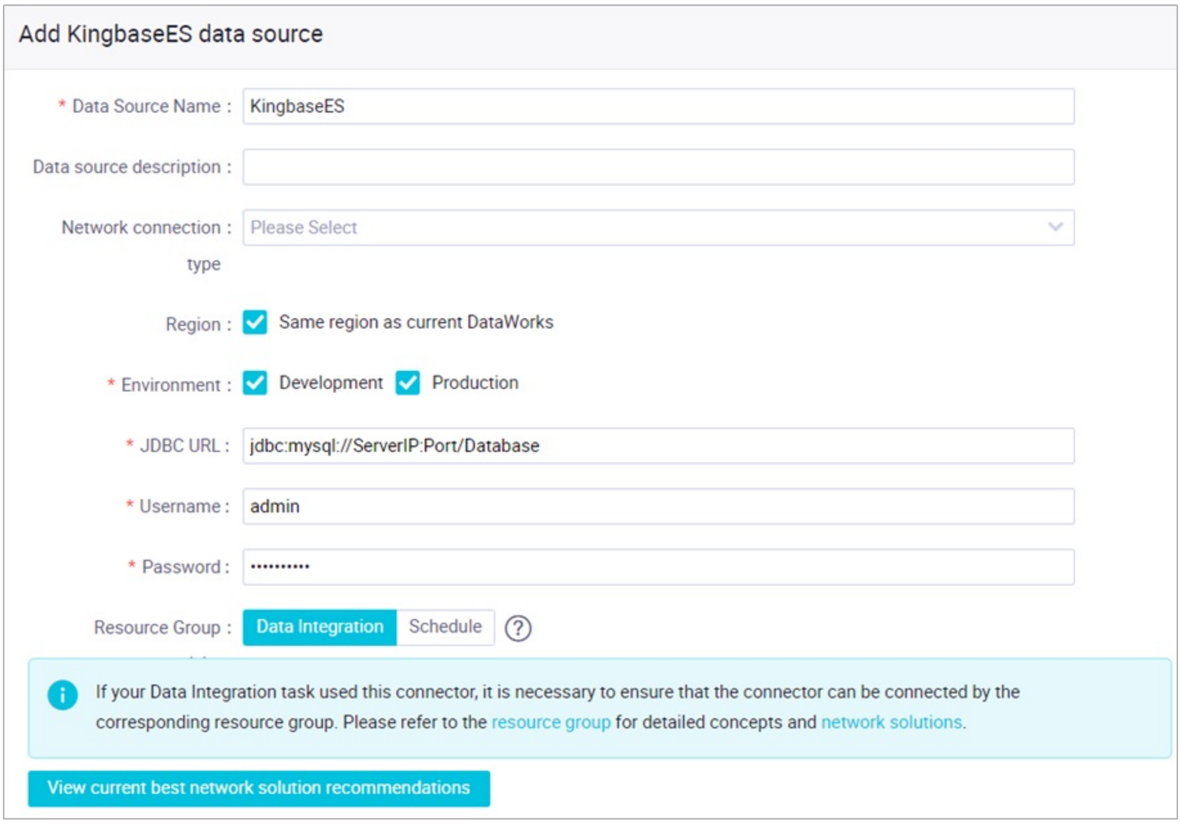
RestAPI data sources support only exclusive resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration.

Procedure

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

If you use a DataWorks workspace in standard mode, perform the following steps to add data sources separately for the development and production environments:

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **KingbaseES** in the Big Data Storage section.
4. In the **Add KingbaseES data source** dialog box, configure the parameters.



Add KingbaseES data source

* Data Source Name : KingbaseES

Data source description :

Network connection type : Please Select

Region : ☑ Same region as current DataWorks

* Environment : ☑ Development ☑ Production

* JDBC URL : jdbc:mysql://ServerIP:Port/Database

* Username : admin

* Password : ••••••••••

Resource Group : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database. Specify this parameter in the format of `jdbc:sap://host:Port?currentschema=SCHEMA`.

Parameter	Description
Username	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

5. Set **Resource Group connectivity** to **Data Integration**.
6. Find the desired resource group in the resource group list in the lower part of the dialog box and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   > **Note** You can use only exclusive resource groups. For more information about exclusive resource groups for Data Integration, see Create and use an exclusive resource group for Data Integration.

7. After the data source passes the connectivity test, click **Complete**.

What's next

After you add a RestAPI data source, you can configure a synchronization node for the data source by using the codeless user interface (UI) or code editor.

5.1.33. Add an ApsaraDB for OceanBase data source

DataWorks provides ApsaraDB for OceanBase Reader and ApsaraDB for OceanBase Writer for you to read data from and write data to ApsaraDB for OceanBase data sources. You can use the codeless user interface (UI) or code editor to configure synchronization nodes for ApsaraDB for OceanBase data sources. This topic describes how to add an ApsaraDB for OceanBase data source.

Context

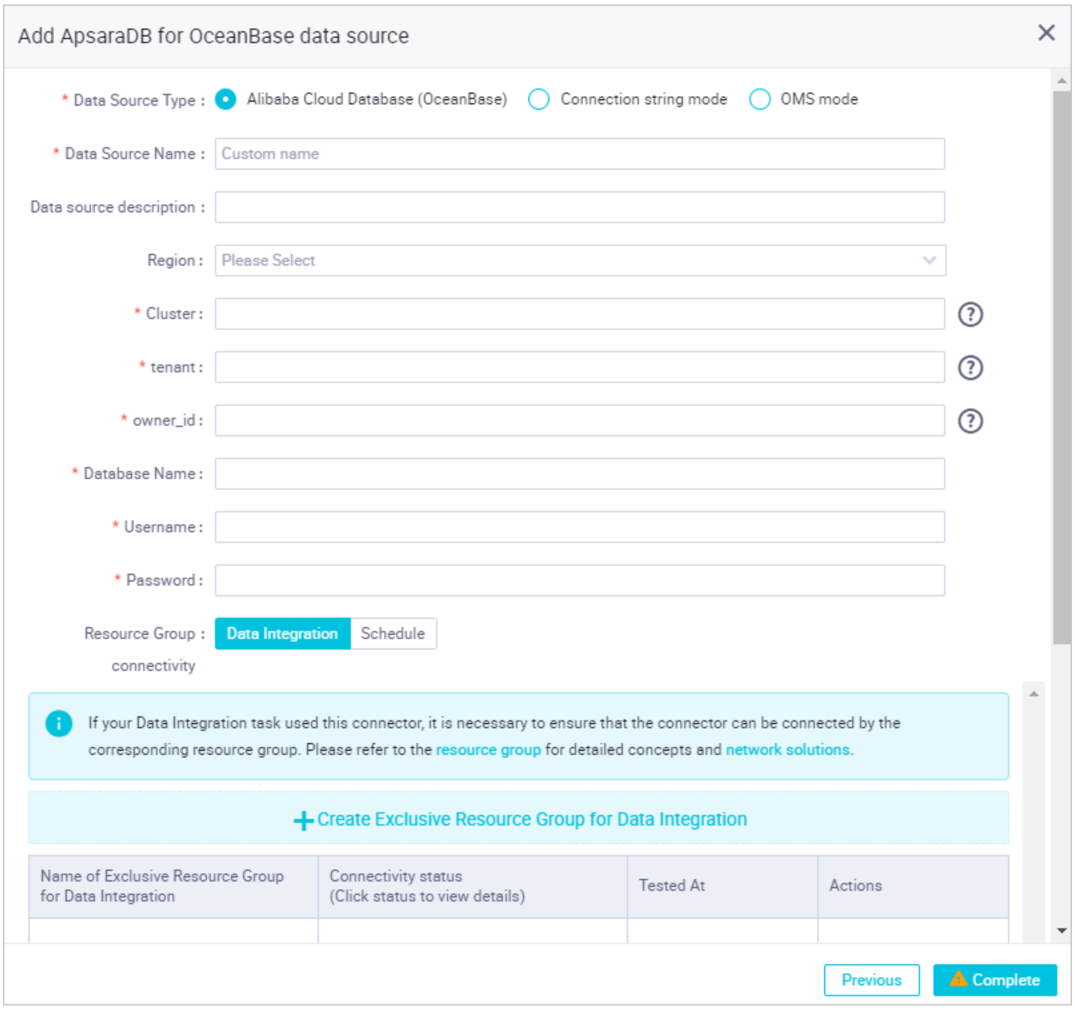
Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Add an ApsaraDB for OceanBase data source

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **ApsaraDB for OceanBase** in the Relational Database section.
4. In the **Add ApsaraDB for OceanBase data source** dialog box, configure the parameters.

   You can use one of the following modes to add an ApsaraDB for OceanBase data source: **Alibaba Cloud Database (OceanBase)** and **Connection string mode**.

   - The following table describes the parameters you must configure if you add an ApsaraDB for OceanBase data source by using the Alibaba Cloud database (OceanBase) mode.



Add ApsaraDB for OceanBase data source

* Data Source Type : ● Alibaba Cloud Database (OceanBase) ○ Connection string mode ○ OMS mode

* Data Source Name : Custom name

Data source description :

Region : Please Select

* Cluster :

* tenant :

* owner_id :

* Database Name :

* Username :

* Password :

Resource Group connectivity : Data Integration | Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

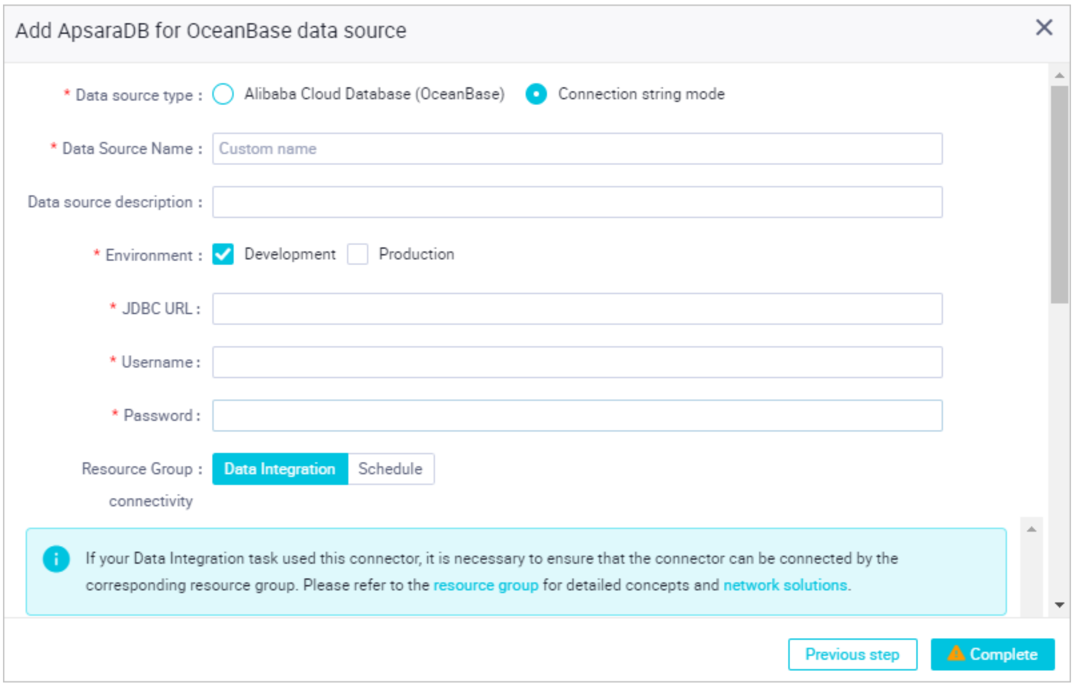
Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Tested At	Actions

Previous | Complete

Parameter	Description
Data source type	The type of the data source. Set this parameter to **Alibaba Cloud Database (OceanBase)**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Region	The region where the data source resides.
Cluster	The ID of the ApsaraDB for OceanBase cluster. You can log on to the ApsaraDB for OceanBase console and view the ID of the cluster on the Basic Information page.
tenant	The ID of the tenant in the ApsaraDB for OceanBase cluster. You can log on to the ApsaraDB for OceanBase console and view the ID of the tenant on the Basic Information page.

Parameter	Description
owner_id	The ID of the Alibaba Cloud account that is used to purchase the ApsaraDB for OceanBase cluster. You can use this account to log on to the ApsaraDB for OceanBase console and view the ID of the account on the **Security Settings** page.
Database Name	The name of the ApsaraDB for OceanBase database.
Username	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

- The following table describes the parameters you must configure if you add an ApsaraDB for OceanBase data source by using the connection string mode.



Add ApsaraDB for OceanBase data source

* Data source type : ○ Alibaba Cloud Database (OceanBase) ● Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* JDBC URL :

* Username :

* Password :

Resource Group connectivity : Data Integration Schedule

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

Parameter	Description
Data source type	The type of the data source. Set this parameter to **Connection string mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
JDBC URL	The Java Database Connectivity (JDBC) URL of the database. Specify this parameter in the format of `jdbc:oceanbase://IP address:Port number/Database`.
Username	The username that is used to connect to the database.
Password	The password that is used to connect to the database.

5. Set **Resource Group connectivity** to **Data Integration**.
6.
7. After the data source passes the connectivity test, click **Complete**.

What's next

You can use the added ApsaraDB for OceanBase data source in your data synchronization node. For more information, see Overview.

5.1.34. Add a Kafka data source

Kafka is a distributed messaging service that is widely used in big data fields such as log collection, monitoring data aggregation, streaming data processing, and online and offline analytics. You can configure synchronization nodes to read data from or write data to Kafka data sources. This topic describes how to add a Kafka data source.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Precautions

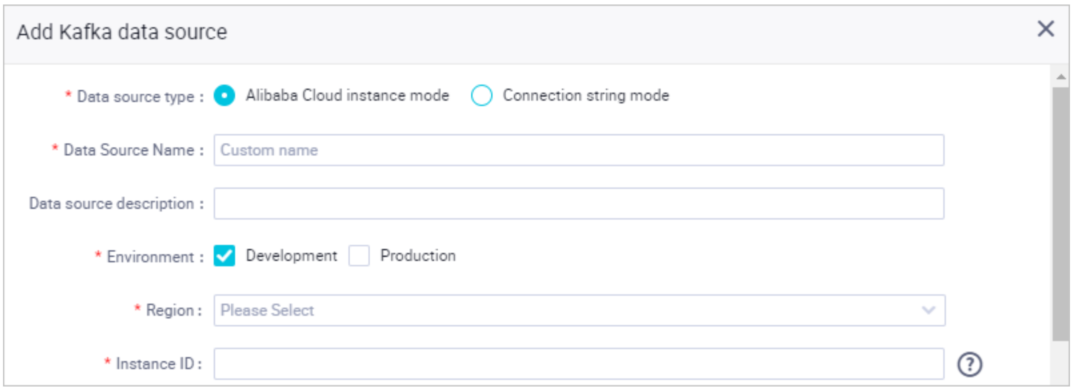
Message Queue for Apache Kafka data sources and self-managed Kafka data sources are supported. However, the versions of self-managed Kafka data sources must range from 0.10.2 to 2.2.x.

Add a data source

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the Message Queue section of the **Add data source** dialog box, click **Kafka**.
4. In the **Add Kafka data source** dialog box, configure the parameters.
   i. Configure basic information for the Kafka data source.

      You can use one of the following modes to add a Kafka data source: **Alibaba Cloud Instance Mode** and **Connection String Mode**.

- The following table describes the parameters you can configure when you add a Kafka data source in **Alibaba Cloud instance mode**.



Add Kafka data source

* Data source type : ● Alibaba Cloud instance mode ○ Connection string mode

* Data Source Name : Custom name

Data source description :

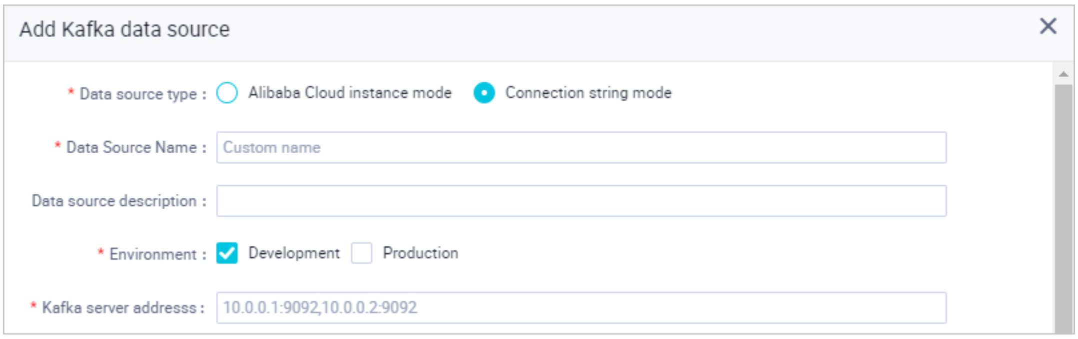
* Environment : ☑ Development □ Production

* Region : Please Select

* Instance ID :

Parameter	Description
Data Source Type	The mode in which the data source is added. Set this parameter to **Alibaba Cloud Instance Mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data Source Description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Region	The region where the Message Queue for Apache Kafka instance resides.
Instance ID	The ID of the Message Queue for Apache Kafka instance. You can log on to the Message Queue for Apache Kafka console and obtain the ID of the instance on the **Instances** page.

- The following table describes the parameters you can configure when you add a Kafka data source in **connection string mode**.



Add Kafka data source

* Data source type : ○ Alibaba Cloud instance mode ● Connection string mode

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

* Kafka server addresss : 10.0.0.1:9092,10.0.0.2:9092

Parameter	Description
Data Source Type	The type of the data source. Set this parameter to **Connection String Mode**.
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data Source Description	The description of the data source. The description can be a maximum of 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is displayed only when the workspace is in standard mode.
Kafka server address	The **address** of a Message Queue for Apache Kafka instance. The address consists of the IP address and port number of a broker. To obtain the **address** of the instance, you can log on to the Message Queue for Apache Kafka console, find the instance on the **Instances** page, and then click the instance name to go to the instance details page. Separate multiple addresses with commas (,), such as `10.0.0.1:9092,10.0.0.2:9092`.

ii. Configure authentication information for the Kafka data source.

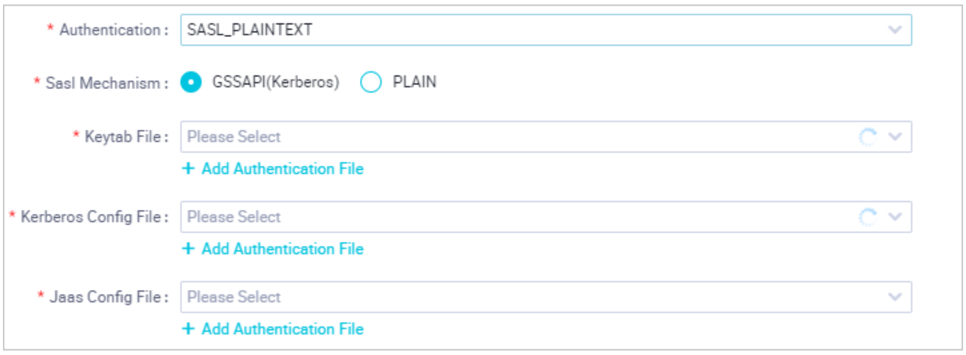
Third-party identity authentication mechanisms are used to perform strict identity authentication on users and services. These mechanisms prevent untrusted applications or services from accessing data and improve the stability of data access during data synchronization. DataWorks provides third-party authentication mechanisms to ensure the data security of Kafka data sources. When you add a Kafka data source, you can set the **Authentication** parameter to one of the following mechanisms: **SASL_PLAINTEXT**, **SASL_SSL**, and **SSL**. This way, only trusted applications and services can access data in the Kafka data source.

Note

- Before you use a third-party authentication mechanism to perform identity authentication, you must upload the required authentication files on the Authentication File Management page of the DataWorks console. For more information, see Upload and reference an authentication file.
- Authentication files are required only for a Kafka data source that is used as the source in a batch synchronization node. If you want to configure authentication and upload authentication files for a Kafka data source that is used as the source in a real-time synchronization node, submit a ticket.
- If you do not need to perform an identity authentication on applications or services, you can set the **Authentication** parameter to **None** when you add the Kafka data source.
- The parameter configurations of third-party authentication for a Kafka data source added in **Alibaba Cloud instance mode** are the same as those of third-party authentication for a Kafka data source added in **connection string mode**.

The following descriptions provide the configurations of the preceding authentication mechanisms:

- **SASL_PLAINTEXT** is a simple authentication mechanism that is implemented based on a username and a password. The following table describes the parameter configurations of the **SASL_PLAINTEXT** mechanism.



* Authentication : SASL_PLAINTEXT

* Sasl Mechanism : ● GSSAPI(Kerberos) ○ PLAIN

* Keytab File : Please Select
+ Add Authentication File

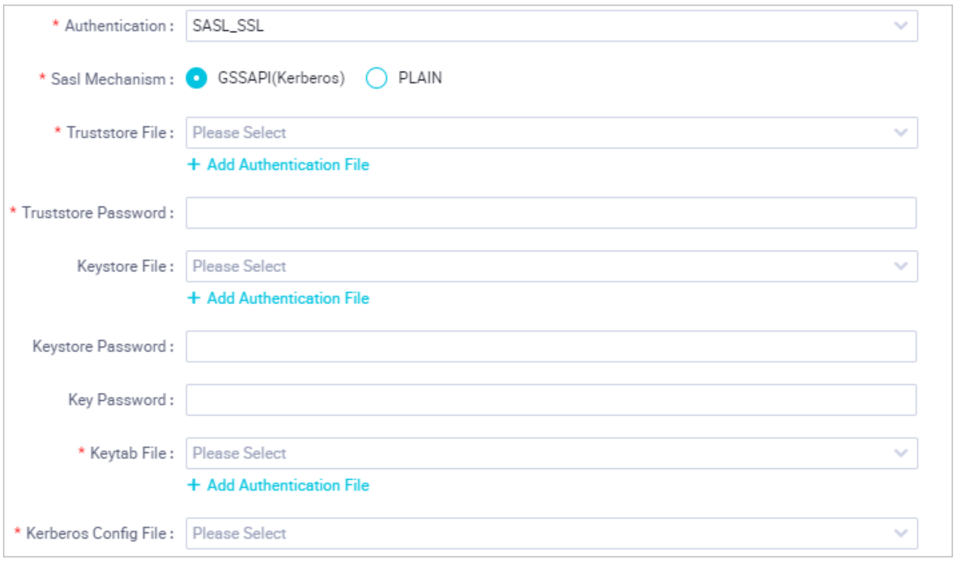
* Kerberos Config File : Please Select
+ Add Authentication File

* Jaas Config File : Please Select
+ Add Authentication File

Parameter	Description
Sasl Mechanism	**GSSAPI(Kerberos)** and **PLAIN** are supported. Both authentication methods use the Simple Authentication and Security Layer (SASL) framework. PLAIN is a simple authentication method that is implemented based on a username and a password.
Keytab File	The keytab file that is used to store the key information of applications and services. The **Jaas Config File** parameter references the keytab file specified in the Keytab File parameter. You can directly reference the keytab file that is uploaded on the Authentication File Management page. You can also click **Add Authentication File** to upload a new keytab file. Note This parameter is required only when the **Sasl Mechanism** parameter is set to **GSSAPI(Kerberos)**.

Parameter	Description
Kerberos Config File	Specifies the configuration file that is used to store the address information of the key distribution center (KDC). This parameter is used to specify the system property `java.security.krb5.conf` for secure authentication. You can directly reference the keytab file that is uploaded on the Authentication File Management page. You can also click **Add Authentication File** to upload a new keytab file. Note This parameter is required only when the **Sasl Mechanism** parameter is set to **GSSAPI(Kerberos)**.
Jaas Config File	Specifies the configuration file that is used to store authentication and authorization information. This parameter is used to specify the system property `java.security.auth.login.config` for secure authentication. You can directly reference the keytab file that is uploaded on the Authentication File Management page. You can also click **Add Authentication File** to upload a new keytab file. Note If you set the **Sasl Mechanism** parameter to **GSSAPI(Kerberos)**, the **Jaas Config File** parameter references the keytab file specified in the **Keytab File** parameter for authentication.

- **SASL_SSL** is a simple mechanism that is used to perform authentication between clients and servers. The following table describes the parameter configurations of the **SASL_SSL** mechanism.



* Authentication : SASL_SSL

* Sasl Mechanism : ● GSSAPI(Kerberos) ○ PLAIN

* Truststore File : Please Select

+ Add Authentication File

* Truststore Password :

Keystore File : Please Select

+ Add Authentication File

Keystore Password :

Key Password :

* Keytab File : Please Select

+ Add Authentication File

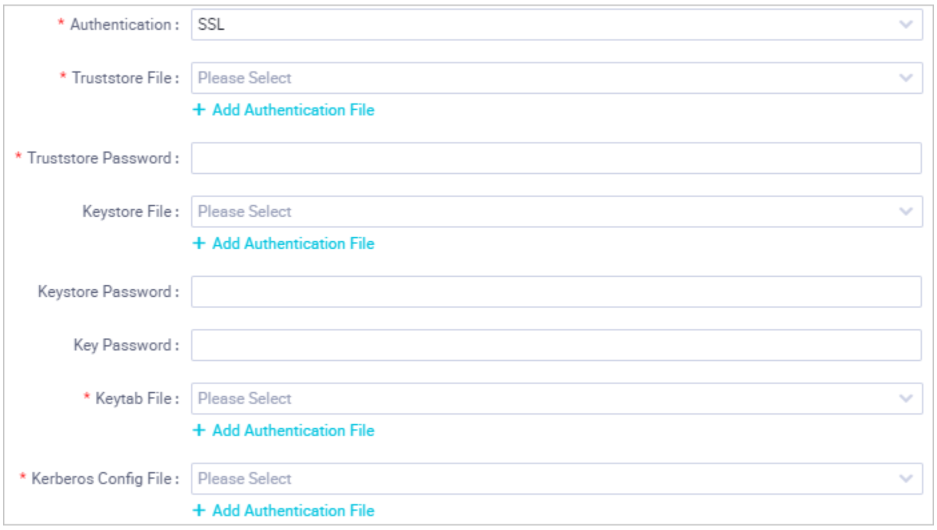
* Kerberos Config File : Please Select

Parameter	Description

Parameter	Description
Sasl Mechanism	**GSSAPI(Kerberos)** and **PLAIN** are supported. Both authentication methods use the Simple Authentication and Security Layer (SASL) framework. PLAIN is a simple authentication method that is implemented based on a username and a password.
Truststore File	Specifies the Truststore file that is used to store the digital certificates provided by Certificate Authority (CA) for the Kafka cluster. These certificates are verified when an application or service accesses the Secure Sockets Layer (SSL) server to ensure that the application or service is trusted. You can directly reference the keytab file that is uploaded on the Authentication File Management page. You can also click **Add Authentication File** to upload a new keytab file. **Note** CA digital certificates are used to check whether access sources are trusted.
Truststore Password	The password that is used to obtain the content of the CA digital certificates of the Kafka cluster.
Keystore File	The Keystore file that is used to store the trusted CA digital certificates and key information of the Kafka cluster. You can directly reference the keytab file that is uploaded on the Authentication File Management page. You can also click **Add Authentication File** to upload a new keytab file.
Keystore Password	The password that is used to access the **Keystore file**.
Key Password	The password that is used to obtain the specified key pair in the **Keystore file**.
Keytab File	The keytab file that is used to store the key information of applications and services. The **Jaas Config File** parameter references the keytab file specified in the Keytab File parameter. You can directly reference the keytab file that is uploaded on the Authentication File Management page. You can also click **Add Authentication File** to upload a new keytab file. **Note** This parameter is required only when the **Sasl Mechanism** parameter is set to **GSSAPI(Kerberos)**.

Parameter	Description
Kerberos Config File	Specifies the configuration file that is used to store the address information of the key distribution center (KDC). This parameter is used to specify the system property `java.security.krb5.conf` for secure authentication. ⓘ **Note**
Jaas Config File	Specifies the configuration file that is used to store authentication and authorization information. This parameter is used to specify the system property `java.security.auth.login.config` for secure authentication. ⓘ **Note** If you set the **Sasl Mechanism** parameter to **GSSAPI(Kerberos)**, the **Jaas Config File** parameter references the keytab file specified in the **Keytab File** parameter for authentication.

- **SSL** is a mechanism that is used to perform authentication between clients and servers. The following table describes the parameter configurations of the **SSL** mechanism.

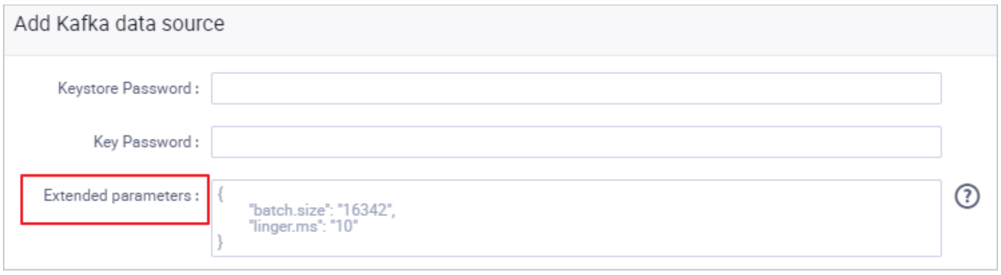


* Authentication :	SSL
* Truststore File :	Please Select + Add Authentication File
* Truststore Password :	
Keystore File :	Please Select + Add Authentication File
Keystore Password :	
Key Password :	
* Keytab File :	Please Select + Add Authentication File
* Kerberos Config File :	Please Select + Add Authentication File

Parameter	Description
Truststore File	Specifies the Truststore file that is used to store the digital certificates provided by Certificate Authority (CA) for the Kafka cluster. These certificates are verified when an application or service accesses the Secure Sockets Layer (SSL) server to ensure that the application or service is trusted. **Note** CA digital certificates are used to check whether access sources are trusted.
Truststore Password	The password that is used to obtain the content of the CA digital certificates of the Kafka cluster.
Keystore File	The Keystore file that is used to store the trusted CA digital certificates and key information of the Kafka cluster.
Keystore Password	The password that is used to access the **Keystore file**.
Key Password	The password that is used to obtain the specified key pair in the **Keystore file**.

5. (Optional)Configure extended parameters for the Kafka data source.

   You can configure extended parameters for the Kafka data source based on your business requirements. The extended parameters are parameters that are related to the producer and consumer of Kafka. Specify the parameters in the JSON format.



Add Kafka data source

Keystore Password :

Key Password :

Extended parameters :

```
{
    "batch.size": "16342",
    "linger.ms": "10"
}
```

You can configure the following parameters:

- batch.size: specifies the buffer size of messages that are sent to each partition. The buffer size indicates the total bytes of messages. In this example, this parameter is set to 16342.
- linger.ms: specifies the maximum storage duration of each message in the buffer. Unit: milliseconds. In this example, this parameter is set to 10.

```
{
"batch.size":"16342",
"linger.ms":"10"
}
```

> Note When you configure a batch synchronization node by using the code editor or a real-time single-table synchronization node, if you set producer- or consumer-related parameters to values that are different from those of the parameters you configured in the **extended parameters**, the values of the parameters that you configured for the synchronization node take effect.

6. Test the network connectivity of the Kafka data source.
   i. Select **Data Integration** for the **Resource Group connectivity** parameter.
   ii. In the resource group list, find the resource group that you want to use and click **Test connectivity** in the Actions column.

      A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

      > Note
      >
      > - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
      > - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

7. After the data source passes the connectivity test, click **Complete**.

What's next

You can use the added Kafka data source in your data synchronization node. For more information, see Overview.

5.1.35. Add a Db2 data source

The Db2 data source provided by Alibaba Cloud DataWorks is a data hub that allows you to read data from or write data to a Db2 database by using Db2 Reader or Db2 Writer. Before you read data from or write data to a Db2 database, you must add the Db2 database to DataWorks. Then, you can use the codeless user interface (UI) or code editor to configure a data sync node to read data from or write data to the Db2 database. This topic describes how to add a Db2 data source.

Prerequisites

A resource group for Data Integration is created to run the data sync node.

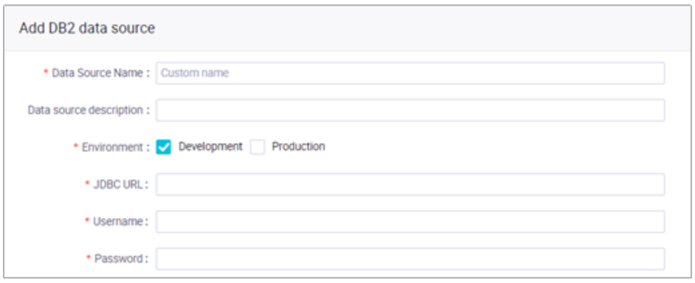
You must use a resource group for Data Integration to run the data sync node. When you add a Db2 data source, you must test the connectivity between the data source and the resource group to ensure that the resource group can connect to the data source. We recommend that you use an exclusive resource group for Data Integration. For more information, see Overview.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Procedure

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.
   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.
2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **DB2** in the Relational Database section.
4. In the **Add DB2 data source** dialog box, set the parameters.



Add DB2 data source

* Data Source Name : Custom name

Data source description :

* Environment : ☑ Development □ Production

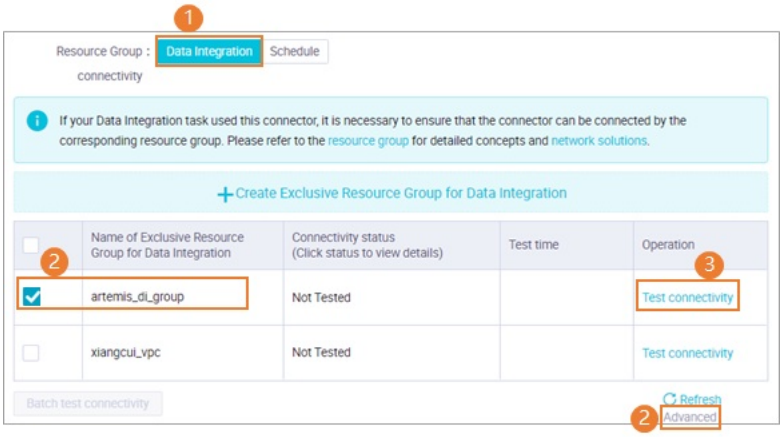
* JDBC URL :

* Username :

* Password :

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data source description	The description of the data source. The description can be up to 80 characters in length.
Environment	The environment in which the data source is used. Valid values: **Development** and **Production**. **Note** This parameter is available only when the workspace is in standard mode.
JDBC URL	The Java Database Connectivity (JDBC) URL of the Db2 database, in the format of `jdbc:db2://ServerIP:Port/Database` .
User name	The username that is used to connect to the Db2 database.
Password	The password that is used to connect to the Db2 database.

5. Test network connectivity.



Resource Group : Data Integration Schedule

connectivity

If your Data Integration task used this connector, it is necessary to ensure that the connector can be connected by the corresponding resource group. Please refer to the resource group for detailed concepts and network solutions.

+Create Exclusive Resource Group for Data Integration

☐	Name of Exclusive Resource Group for Data Integration	Connectivity status (Click status to view details)	Test time	Operation
☑	artemis_di_group	Not Tested		Test connectivity
☐	xiangcui_vpc	Not Tested		Test connectivity

Batch test connectivity

Refresh

Advanced

i. Click the **Data Integration** tab.

ii. Select the resource group that you want to use.

- We recommend that you use an exclusive resource group for Data Integration. If you do not have an exclusive resource group for Data Integration, you can click **Create Exclusive Resource Group for Data Integration** to purchase one by following the on-screen instructions.
- If you want to use a custom resource group, click **Advanced** to view these resource groups and select the resource group that you want to use.

iii. Find the resource group that you want to use and click **Test connectivity** in the Actions column.

A sync node uses only one resource group for Data Integration. If you want to test the network connectivity between multiple types of resource groups and the data source, you can select the resource groups and click **Batch test connectivity**.

6. After the data source passes the connectivity test, click **Complete**.

What's next

After the Db2 data source is added, you can configure a data sync node to read data from or write data to the data source. Before you configure the data sync node, you must learn how to configure Db2 Reader or Db2 Writer that will be used by the data sync node. For more information, see DB2 Reader and DB2 Writer.

5.1.36. Add an Amazon S3 data source

Amazon Simple Storage Service (Amazon S3) is an object storage service that allows you to store and retrieve any amount of data from anywhere. You can add Amazon S3 data sources to your DataWorks workspace and then read data from and write data to the added data sources. This topic describes how to add an Amazon S3 data source.

Prerequisites

A resource group for Data Integration is created to run the synchronization node.

You must use a resource group for Data Integration to run the synchronization node. When you add an Amazon S3 data source, you must test the connectivity between the data source and the resource group to ensure that the data source is connected to the resource group. You must use an exclusive resource group for Data Integration. For more information, see Overview.

Context

Workspaces in standard mode support the data source isolation feature. You can add data sources separately for the development and production environments to isolate the data sources. This helps keep your data secure. For more information, see Isolate connections between the development and production environments.

Limits

Amazon S3 data sources in the Chinese mainland and Hong Kong (China) are not supported.

Add an Amazon S3 data source

1. Go to the **Data Source** page.

   i. Log on to the DataWorks console.

   ii. In the left-side navigation pane, click **Workspaces**.

   iii. After you select the region where the required workspace resides, find the workspace and click **Data Integration** in the Actions column.

   iv. In the left-side navigation pane of the Data Integration page, choose **Data Source > Data Sources** to go to the Data Source page.

2. On the **Data Source** page, click **Add data source** in the upper-right corner.
3. In the **Add data source** dialog box, click **S3** in the Semi-structuredstorage section.
4. In the **Add S3 data source** dialog box, set the parameters as required.

   i. Configure the basic information of the Amazon S3 data source.

Parameter	Description
Data Source Name	The name of the data source. The name can contain letters, digits, and underscores (_) and must start with a letter.
Data Source Description	The description of the data source. The description can be up to 80 characters in length.
Endpoint	The endpoint of the Amazon S3 data source. Example: `http://s3.ap-northeast-1.amazonaws.com` . You can query the endpoint of the source Amazon S3 bucket in the Amazon S3 console.
Bucket	The name of the Amazon S3 bucket. A bucket is a storage space that serves as a container for storing objects. You can create one or more buckets and add one or more objects to each bucket. During data synchronization, DataWorks can search for objects only in the bucket that is specified by this parameter.
AccessKey ID	The AccessKey ID of the account that you use to connect to the Amazon S3 bucket.
AceessKey Secret	The AccessKey secret of the account that you use to connect to the Amazon S3 bucket.

5. Test the network connectivity between the Amazon S3 data source and the resource group.

   i. Set the **Resource Group connectivity** parameter to **Data Integration**.

   ii. In the resource group list, find the resource group that you want to use and click **Test connectivity** in the Actions column.

   A synchronization node can use only one type of resource group. To ensure that your synchronization nodes can be normally run, you must test the connectivity of all the resource groups for Data Integration on which your synchronization nodes will be run. If you want to test the connectivity of multiple resource groups for Data Integration at a time, select the resource groups and click **Batch test connectivity**. For more information, see Select a network connectivity solution.

   Note

   - By default, the resource group list displays only exclusive resource groups for Data Integration. To ensure the stability and performance of data synchronization, we recommend that you use exclusive resource groups for Data Integration.
   - If you want to test the network connectivity between the shared resource group or a custom resource group and the data source, click **Advanced** below the resource group list. In the **Warning** message, click **Confirm**. Then, all available shared and custom resource groups appear in the resource group list.

6. After the data source passes the connectivity test, click **Complete**.

What's next

You have learned how to add an Amazon S3 data source. You can proceed to subsequent tutorials. In subsequent tutorials, you will learn how to configure Amazon S3 Reader. For more information, see Amazon S3 Reader.

5.2. Configure a reader

5.2.1. DRDS Reader

DRDS Reader reads data from Distributed Relational Database Service (DRDS). This topic describes the data types and parameters that are supported by DRDS Reader and how to configure DRDS Reader by using the codeless user interface (UI) and code editor.

Background information

DRDS Reader supports only MySQL engines. DRDS is a distributed MySQL database service. Most of the communication protocols that DRDS uses are the same as those used by MySQL.

Notice Only exclusive resource groups for Data Integration can be used to read data from DRDS instances that run MySQL 8.0.

DRDS Reader connects to a remote DRDS database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the statement on the database and returns data. Then, DRDS Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

DRDS Reader generates the SQL statement based on the settings of the table, column, and where parameters, and sends the generated statement to the DRDS database. DRDS does not support MySQL specifications, such as JOIN statements.

Data types

DRDS Reader supports most DRDS data types. Make sure that the data types of your database are supported.

The following table describes the data types that are supported by DRDS Reader.

Category	DRDS data type
Integer	INT, TINYINT, SMALLINT, MEDIUMINT, and BIGINT
Floating point	FLOAT, DOUBLE, and DECIMAL
String	VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT
Date and time	DATE, DATETIME, TIMESTAMP, TIME, and YEAR
Boolean	BIT and BOOLEAN
Binary	TINYBLOB, MEDIUMBLOB, BLOB, LONGBLOB, and VARBINARY

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table from which you want to read data.	Yes	No default value

Parameter	Description	Required	Default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. The default value is [*], which indicates all the columns in the source table. ● You can select specific columns to read. ● The column order can be changed. You can configure DRDS Reader to read the specified columns in an order different from that specified by the schema of the table. ● Constants are supported. The column names must be arranged in compliance with the SQL syntax supported by MySQL, such as ``["id", "`table`", "1", "'bazhen.csy'", "null", "to_char(a + 1)", "2.3", "true"]`` . Description of the parameters in the preceding example: ○ id: a column name. ○ table: the name of a column that contains reserved keywords. ○ 1: an integer constant. ○ bazhen.csy: a string constant. ○ null: a null pointer. ○ to_char(a + 1): a function expression that is used to calculate the length of a string. ○ 2.3: a floating-point constant. ○ true: a Boolean value. ● The column parameter must explicitly specify all the columns from which you want to read data. The parameter cannot be left empty.	Yes	No default value
where	The WHERE clause. DRDS Reader generates an SQL statement based on the settings of the column, table, and where parameters, and uses the generated statement to read data. ● You can use the WHERE clause to read incremental data. ● If the where parameter is not specified or is left empty, DRDS Reader reads full data. For example, you can set this parameter to `STRTODATE('${bdp.system.bizdate}','%Y%m%d') <= taday AND taday < DATEADD(STRTODATE('${bdp.system.bizdate}', '%Y%m%d'), interval 1 day)` to read the data that is generated on the current day.	No	No default value

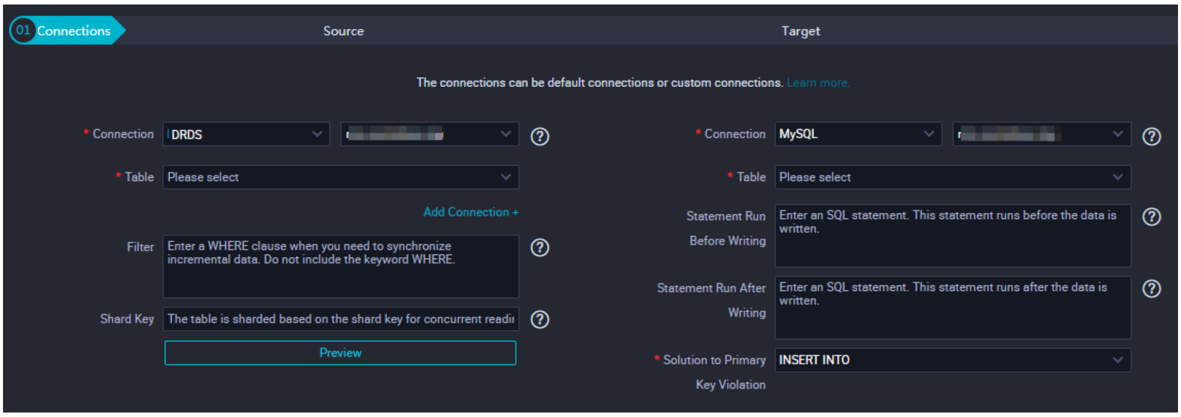
Configure DRDS Reader by using the codeless UI

Create a synchronization node and configure the node. For more information, see Configure a synchronization node by using the codeless UI.

Perform the following steps on the configuration tab of the synchronization node:

1. Configure data sources.

Set parameters in the **Source** and **Target** sections for the synchronization node.



01 Connections

Source

Target

The connections can be default connections or custom connections. Learn more.

Source:

- * Connection: DRDS [redacted]
- * Table: Please select
- Add Connection +
- Filter: Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.
- Shard Key: The table is sharded based on the shard key for concurrent readi
- Preview

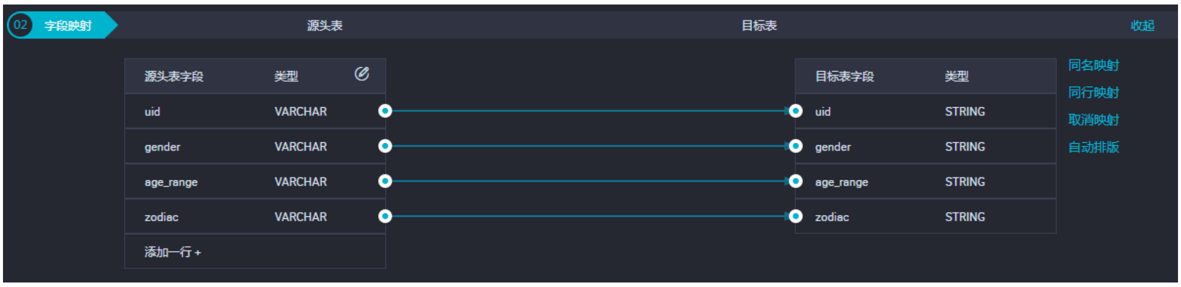
Target:

- * Connection: MySQL [redacted]
- * Table: Please select
- Statement Run Before Writing: Enter an SQL statement. This statement runs before the data is written.
- Statement Run After Writing: Enter an SQL statement. This statement runs after the data is written.
- * Solution to Primary Key Violation: INSERT INTO

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table from which you want to read data. This parameter is equivalent to the table parameter that is described in the preceding section.
Filter	The condition that is used to filter the data you want to read. Filtering based on the LIMIT keyword is not supported. The SQL syntax is determined by the selected data source.
Shard Key	The shard key. You can use a column in the source table as the shard key. We recommend that you use the primary key column or an indexed column. Only integer columns are supported. If you set this parameter, data sharding is performed based on the value of this parameter, and parallel threads can be used to read data. This improves data synchronization efficiency. **Note** The Shard Key parameter is displayed only after you select the data source for the synchronization node.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



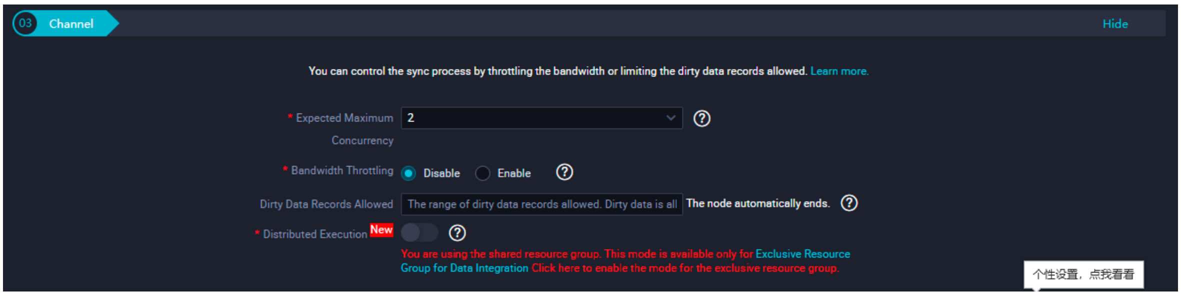
源头表字段	类型
uid	VARCHAR
gender	VARCHAR
age_range	VARCHAR
zodiac	VARCHAR

目标表字段	类型
uid	STRING
gender	STRING
age_range	STRING
zodiac	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

Operation	Description
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source and destination tables. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	Click Add to add a field. Take note of the following rules when you add a field: ◦ You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ◦ You can use scheduling parameters, such as ${bizdate}. ◦ You can enter functions that are supported by relational databases, such as now() and count(1). ◦ If the field that you entered cannot be parsed, the value of Type for the field is **Unidentified**.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure DRDS Reader by using the code editor

In the following code, a synchronization node is configured to read data from a DRDS database. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"drds",// The reader type.
            "parameter":{
                "datasource":"",// The name of the data source.
                "column":[// The names of the columns from which you want to read data.
                    "id",
                    "name"
                ],
                "where":"",// The WHERE clause.
                "table":"",// The name of the table from which you want to read data.
                "splitPk": ""// The shard key.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",// The writer type.
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1,// The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}:"Writer"
            }
        ]
    }
}
```

Additional information

- Consistent view

  DRDS cannot provide a consistent view of multiple tables in multiple databases because it is a distributed database service. DRDS Reader obtains different snapshots from different table shards, but cannot obtain the snapshot of table shards at the same time slice. As a result, DRDS Reader cannot ensure strong consistency for data queries.

- Character encoding

  DRDS supports flexible encoding configurations. You can specify the encoding format for an entire instance and specific fields, tables, and databases. The configurations at the field, table, database, and instance levels are prioritized in descending order. We recommend that you use UTF-8 for a database.

DRDS Reader uses JDBC to read data. This enables DRDS Reader to automatically convert the encoding formats of characters. Therefore, you do not need to specify an encoding format when you use DRDS Reader.

If you specify an encoding format but data in the source DRDS database is written in a different encoding format, DRDS Reader cannot recognize this inconsistency and may export garbled characters.

- Incremental data synchronization

  DRDS Reader uses JDBC to connect to a database and uses a SELECT statement with a `WHERE` clause to read incremental data.

  - For batch data, incremental add, update, and remove operations (including logically remove operations) are distinguished by timestamps. Specify the WHERE clause based on a specific timestamp. The time indicated by the timestamp must be later than the time indicated by the latest timestamp in the previous synchronization.
  - For streaming data, specify the WHERE clause based on the ID of a specific record. The ID must be greater than the maximum ID involved in the previous synchronization.

  If the data that is added or modified cannot be distinguished, DRDS Reader can read only full data.

- You cannot configure filter conditions for physical tables in the WHERE clause.

5.2.2. HBase Reader

HBase Reader reads data from HBase. This topic describes the data types and parameters that are supported by HBase Reader and how to configure HBase Reader by using the codeless user interface (UI) and code editor.

HBase Reader connects to a remote HBase database by using a Java client of HBase, scans and reads data based on a specific rowkey range, assembles the data into abstract datasets of the data types supported by Data Integration, and then sends the datasets to a writer.

Limits

HBase Reader supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.

Supported features

- HBase Reader can read data from HBase 0.94.X, HBase 1.1.X, and HBase 2.X.
  - If you use HBase 0.94.X, set the plugin parameter to 094x.

    ```
    "reader": {
            "plugin": "094x"
        }
    ```

  - If you use HBase 1.1.X or HBase 2.X, set the plugin parameter to 11x.

    ```
    "reader": {
            "plugin": "11x"
        }
    ```

    Note HBase 1.1.X Reader is compatible with HBase 2.0. If you have questions when you use HBase Reader, submit a ticket.

- HBase Reader supports normal and multiVersionFixedColumn modes.

- In normal mode, HBase Reader reads only the latest version of data from an HBase table and converts the data to a two-dimensional table (wide table).

  ```
  hbase(main):017:0> scan 'users'
  ROW                                   COLUMN+CELL
  lisi                                 column=address:city, timestamp=1457101972764, value=beijing
  lisi                                 column=address:contry, timestamp=1457102773908, value=china
  lisi                                 column=address:province, timestamp=1457101972736, value=beijing
  lisi                                 column=info:age, timestamp=1457101972548, value=27
  lisi                                 column=info:birthday, timestamp=1457101972604, value=1987-06-17
  lisi                                 column=info:company, timestamp=1457101972653, value=baidu
  xiaoming                             column=address:city, timestamp=1457082196082, value=hangzhou
  xiaoming                             column=address:contry, timestamp=1457082195729, value=china
  xiaoming                             column=address:province, timestamp=1457082195773, value=zhejiang
  xiaoming                             column=info:age, timestamp=1457082218735, value=29
  xiaoming                             column=info:birthday, timestamp=1457082186830, value=1987-06-17
  xiaoming                             column=info:company, timestamp=1457082189826, value=alibaba
  2 row(s) in 0.0580 seconds }
  ```

  HBase Reader converts the data that is read from the HBase table to the following table.

rowKey	address:city	address:country	address:province	info:age	info:birthday	info:company
lisi	beijing	china	beijing	27	1987-06-17	baidu
xiaoming	hangzhou	china	zhejiang	29	1987-06-17	alibaba

- In multiVersionFixedColumn mode, HBase Reader reads data from an HBase table and converts the data to a narrow table. The narrow table contains four columns rowKey, family:qualifier, timestamp, and value. Before you use HBase Reader to read data, you must specify the columns from which you want to read data. When HBase Reader reads data, it converts each cell in each version of the table to a data record.

```
hbase(main):018:0> scan 'users',{VERSIONS=>5}
ROW                                    COLUMN+CELL
lisi                                  column=address:city, timestamp=1457101972764, value=beijing
lisi                                  column=address:contry, timestamp=1457102773908, value=china
lisi                                  column=address:province, timestamp=1457101972736, value=beijing
lisi                                  column=info:age, timestamp=1457101972548, value=27
lisi                                  column=info:birthday, timestamp=1457101972604, value=1987-06-17
lisi                                  column=info:company, timestamp=1457101972653, value=baidu
xiaoming                              column=address:city, timestamp=1457082196082, value=hangzhou
xiaoming                              column=address:contry, timestamp=1457082195729, value=china
xiaoming                              column=address:province, timestamp=1457082195773, value=zhejiang
xiaoming                              column=info:age, timestamp=1457082218735, value=29
xiaoming                              column=info:age, timestamp=1457082178630, value=24
xiaoming                              column=info:birthday, timestamp=1457082186830, value=1987-06-17
xiaoming                              column=info:company, timestamp=1457082189826, value=alibaba
2 row(s) in 0.0260 seconds }
```

HBase Reader converts the data that is read from the HBase table to the following table.

rowKey	column:qualifier	timestamp	value
lisi	address:city	1457101972764	beijing
lisi	address:country	1457102773908	china
lisi	address:province	1457101972736	beijing
lisi	info:age	1457101972548	27
lisi	info:birthday	1457101972604	1987-06-17
lisi	info:company	1457101972653	beijing
xiaoming	address:city	1457082196082	hangzhou
xiaoming	address:country	1457082195729	china
xiaoming	address:province	1457082195773	zhejiang
xiaoming	info:age	1457082218735	29
xiaoming	info:age	1457082178630	24
xiaoming	info:birthday	1457082186830	1987-06-17
xiaoming	info:company	1457082189826	alibaba

Data types

The following table lists the data types that are supported by HBase Reader.

Category	Data Integration data type	HBase data type
Integer	LONG	SHORT, INT, and LONG
Floating point	DOUBLE	FLOAT and DOUBLE
String	STRING	BINARY_STRING and STRING
Date and time	DATE	DATE

Category	Data Integration data type	HBase data type
Byte	BYTES	BYTES
Boolean	BOOLEAN	BOOLEAN

Parameters

Parameter	Description	Required	Default value
haveKerberos	Specifies whether Kerberos authentication is required. Valid values: true and false. Note • If you set this parameter to true, Kerberos authentication is required, and you must configure the following parameters that are related to Kerberos authentication: ○ kerberosKeytabFilePath ○ kerberosPrincipal ○ hbaseMasterKerberosPrincipal ○ hbaseRegionserverKerberosPrincipal ○ hbaseRpcProtection • If you set this parameter to false, Kerberos authentication is not required, and you do not need to configure the preceding parameters.	No	*false*
hbaseConfig	The properties of the HBase cluster, in the JSON format. The hbase.zookeeper.quorum parameter is required. It specifies the ZooKeeper address of the HBase cluster. You can also configure other properties, such as those related to the cache and batch for scan operations. Note You must use an internal endpoint to access an ApsaraDB for HBase database.	Yes	No default value
mode	The mode in which HBase Reader reads data from HBase. Valid values: normal and multiVersionFixedColumn.	Yes	No default value
table	The name of the HBase table from which you want to read data. The name is case-sensitive.	Yes	No default value
encoding	The encoding format that is used to convert binary data in the HBase byte[] format to strings. Valid values: utf-8 and gbk.	No	*utf-8*

Parameter	Description	Required	Default value

column	The names of the columns from which you want to read data.
	- In normal mode:
	The name parameter specifies the name of the column from which you want to read data. Specify the column in the columnFamily:columnName format, except for the rowkey column. The type parameter specifies the source data type. The format parameter specifies the date format. The value parameter specifies the column value if the column is a constant column. When HBase reader reads data, it does not read data from the constant column, but uses the settings of the value parameter. The following code provides an example:
	`"column":`
	`[`
	`{`
	`  "name": "rowkey",`
	`  "type": "string"`
	`},`
	`{`
	`  "value": "test",`
	`  "type": "string"`
	`}`
	`]`
	In the column parameter, you must specify the type parameter and specify either the name or value parameter.
	- In multiVersionFixedColumn mode:
	The name parameter specifies the name of the column from which you want to read data. Specify the column in the columnFamily:columnName format, except for the rowkey column. The type parameter specifies the source data type. The format parameter specifies the date format. Constant columns are not supported in multiVersionFixedColumn mode. The following code provides an example:
	`"column":`
	`[`
	`{`
	`  "name": "rowkey",`
	`  "type": "string"`
	`},`
	`{`
	`  "name": "info:age",`
	`  "type": "string"`
	`}`
	`]`
		Yes	No default value

maxVersion	The number of versions that are read by HBase Reader when multiple versions are available. Valid values: -1 and integers greater than 1. The value -1 indicates that all versions are read.	Required in multiVersionFixedColumn mode	No default value

Parameter	Description	Required	Default value

range	The rowkey range based on which HBase Reader reads data.
	- startRowkey: the start rowkey.
	- endRowkey: the end rowkey.
	- isBinaryRowkey: the method that is used to convert the specified start and end rowkeys to the byte[] format. Default value: *false*. If you set this parameter to true, the `Bytes.toBytesBinary(rowkey)` method is used. If you set this parameter to false, the `Bytes.toBytes(rowkey)` method is used. The following code provides an example:
	`"range": {`
	`"startRowkey": "aaa",`
	`"endRowkey": "ccc",`
	`"isBinaryRowkey":false`
	`}`
		No	No default value

scanCacheSize	The number of rows that HBase Reader reads from the HBase table each time.	No	*256*
scanBatchSize	The number of columns that HBase Reader reads from the HBase table each time.	No	*100*

Configure HBase Reader by using the codeless UI

This method is not supported.

Configure HBase Reader by using the code editor

In the following code, a synchronization node is configured to read data from HBase in normal mode. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"hbase",// The reader type.
            "parameter":{
                "mode":"normal",// The mode in which HBase Reader reads data. Valid values: normal and multiVer
sionFixedColumn.
                "scanCacheSize":"256",// The number of rows that HBase Reader reads from the HBase table each t
ime.
                "scanBatchSize":"100",// The number of columns that HBase Reader reads from the HBase table eac
h time.
                "hbaseVersion":"094x/11x",// The HBase version.
                "column":[// The names of the columns from which you want to read data.
                    {
                        "name":"rowkey",// The name of a column.
                        "type":"string"// The data type.
                    },
                    {
                        "name":"columnFamilyName1:columnName1",
                        "type":"string"
                    },
                    {
                        "name":"columnFamilyName2:columnName2",
                        "format":"yyyy-MM-dd",
                        "type":"date"
                    },
                    {
                        "name":"columnFamilyName3:columnName3",
                        "type":"long"
                    }
                ],
                "range":{// The rowkey range based on which HBase Reader reads data.
                    "endRowkey":"",// The end rowkey.
                    "isBinaryRowkey":true,// The method that is used to convert the specified start and end row
keys to the byte[] format. Default value: false.
                    "startRowkey":""// The start rowkey.
                },
                "maxVersion":"",// The number of versions that are read by HBase Reader when multiple versions
are available.
                "encoding":"UTF-8",// The encoding format.
                "table":"",// The name of the table from which you want to read data.
                "hbaseConfig":{// The properties of the HBase cluster, in the JSON format.
                    "hbase.zookeeper.quorum":"hostname",
                    "hbase.rootdir":"hdfs://ip:port/database",
                    "hbase.cluster.distributed":"true"
                }
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1,// The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.2.3. HBase20xsql Reader

HBase20xsql Reader reads data from Phoenix tables that are mapped to HBase SQL tables. This topic describes the data types and parameters that are supported by HBase20xsql Reader and how to configure HBase20xsql Reader by using the code editor.

Prerequisites

Before you configure HBase20xsql Reader, you must configure an HBase data source. For more information, see Add an HBase data source.

How it works

HBase20xsql Reader connects to the query server of Phoenix by using the Phoenix thin client, generates an SQL statement based on your configurations, and then sends the statement to the query server. The query server executes the statement to read data from the HBase data source and returns the obtained data to HBase20xsql Reader. Then, HBase20xsql Reader assembles the data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

Limits

- Only HBase2.x data sources and Phoenix 5.x are supported if you run synchronization nodes that use HBase20xsql Reader to synchronize data.
- You can shard a table based on a single column in the table, and the column must be the primary key column of the table.
- If you want to evenly shard a table based on the number of parallel threads, the values in the column that is used for table sharding must be of an integer or string data type.
- The table name, schema name, and column names for an HBase table are case-sensitive and must be in the same case as the table name, schema name, and column names for the mapped Phoenix table.
- HBase20xsql Reader can read data only by using the Phoenix query server. You can use HBase20xsql Reader only after you start the Phoenix query server in your Phoenix service.

Data types

HBase20xsql Reader supports most Phoenix data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by HBase20xsql Reader.

Data Integration data type	Phoenix data type
long	INTEGER, TINYINT, SMALLINT, and BIGINT
double	FLOAT, DECIMAL, and DOUBLE
string	CHAR and VARCHAR
date	DATE, TIME, and TIMESTAMP
bytes	BINARY and VARBINARY
boolean	BOOLEAN

Parameters

Parameter	Description	Required	Default value
queryServerAddress	The address of the Phoenix query server. If you use ApsaraDB for HBase Performance-enhanced Edition (Lindorm) and you want to pass through the user and password parameters, you can append the settings of these parameters to the value of the queryServerAddress parameter. Example: `http://127.0.0.1:8765;user=root;password=root` .	Yes	No default value
serialization	The serialization protocol used by the Phoenix query server.	No	PROTOBUF
table	The name of the table from which you want to read data. The name is case-sensitive.	Yes	No default value
schema	The schema of the table.	No	No default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. If you leave this parameter empty, all columns in the source table are read. This parameter is empty by default.	No	Empty string

Parameter	Description	Required	Default value
splitKey	The column that is used for table sharding when HBase20xsql Reader reads data. If you configure this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This improves data synchronization efficiency. You can use one of the following methods to shard a table. If the splitPoints parameter is left empty, table sharding is performed by using Method 1. • Method 1: Find the maximum value and minimum value in the column specified by the splitKey parameter and evenly shard the table based on the value of the concurrent parameter. Note You can shard a table based on a column in which values are of an integer or string data type. • Method 2: Shard a table based on the value of the splitPoints parameter. Then, the data is synchronized by using the parallel threads specified by the concurrent parameter.	Yes	No default value
splitPoints	The sharding point. If you shard a table based on the maximum value and minimum value of the column that is used for table sharding, data may be intensively distributed to specific regions. We recommend that you specify a value for the splitPoints parameter based on the start key and end key of a region to ensure that a query statement is used to query data only in a region obtained after the table sharding.	No	No default value
where	The WHERE clause. You can configure this parameter to filter data in the source table. HBase20xsql Reader generates an SQL statement based on the settings of the column, table, and where parameters and uses the generated statement to read data.	No	No default value
querySql	The SQL statement that is used for refined data filtering. If you configure the querySql parameter and the queryserverAddress parameter that is required, HBase20xsql Reader ignores the column, table, where, and splitKey parameters that you configured and uses the setting of this parameter for data filtering.	No	No default value

Configure HBase20xsql Reader by using the codeless UI

This method is not supported.

Configure HBase20xsql Reader by using the code editor

In the following code, a synchronization node is configured to read data from HBase. For more information, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"hbase20xsql",// The reader type.
            "parameter":{
                "queryServerAddress": "http://127.0.0.1:8765",  // The address of the Phoenix query server.
                "serialization": "PROTOBUF",  // The serialization protocol used by the Phoenix query server.
                "table": "TEST",    // The name of the table from which you want to read data.
                "column": ["ID", "NAME"],   // The names of the columns from which you want to read data.
                "splitKey": "ID"    // The column that is used for table sharding when HBase20xsql Reader reads
data. The column must be the primary key column of the source table.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1,// The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.2.4. HDFS Reader

HDFS Reader can read data from files stored in Hadoop Distributed File System (HDFS). After HDFS Reader obtains data, it converts the data from the original data types to the data types supported by Data Integration and sends the converted data to a writer.

Notice You can use only an exclusive resource group for Data Integration to run a data synchronization node that uses HDFS Reader. For more information about exclusive resource groups for Data Integration, see Create and use an exclusive resource group for Data Integration.

Background information

HDFS Reader reads data from files in HDFS and converts the data from the original data types to the data types supported by Data Integration.

By default, if HDFS is used as the storage of Hive tables, the Hive tables are stored in HDFS as text files that are not compressed. HDFS Reader reads data in a similar way as OSS Reader.

The Optimized Row Columnar (ORC) file format is an optimized RC file format and allows you to store Hive data in an efficient manner. HDFS Reader uses the OrcSerde class provided by Hive to parse and read data in ORC files.

Take note of the following items when you use HDFS Reader:

- Complex network connections are required between the shared resource group and HDFS. Therefore, we recommend that you use an exclusive resource group for Data Integration to run your synchronization node. Make sure that your exclusive resource group for Data Integration can access the NameNode and DataNode nodes of HDFS.
- By default, HDFS uses a network whitelist to ensure data security. In this case, we recommend that you use exclusive resource groups for Data Integration to run synchronization nodes that use HDFS Reader.
- If you use the code editor to configure a synchronization node that uses HDFS Reader, the network connectivity test for the HDFS data source that you use is not required. If the system reports an error for the connectivity test, you can ignore the error.
- You must use an administrator account to start your synchronization node. Make sure that your administrator account has the permissions to read data from and write data to related HDFS files.

Features

HDFS Reader supports the following features:

- Supports the text, ORC, RC, Sequence, CSV, and Parquet file formats. Data stored in the files in these formats must be organized as logical two-dimensional tables.
- Reads data of various types as strings. Supports constants and column pruning.
- Supports recursive reading and regular expressions that contain asterisks (`*`) and question marks (`?`).
- Compresses ORC files in the Snappy or ZLIB format.
- Compresses SequenceFile files in LZO format.
- Uses multiple threads to read files.
- Compresses CSV files in the GZIP, BZ2, ZIP, LZO, LZO_DEFLATE, or Snappy format.
- Supports Hive 1.1.1 and Hadoop 2.7.1 that works with JDK 1.6. HDFS Reader can normally run with Hive 1.2.0 and Hadoop 2.5.0 or Hadoop 2.6.0 during testing.

Notice HDFS Reader cannot use parallel threads to read a single file due to the internal sharding method.

Data types

Hive maintains the metadata of files and stores the metadata in its own metadatabase, such as a MySQL database. HDFS Reader cannot access or query the metadata in the metadatabase of Hive. Therefore, you must specify the data types that you want to convert.

The following table describes the mappings between data types in RC, Parquet, ORC, text, and Sequence files in Hive and the data types supported by Data Integration.

Category	Data Integration data type	Hive data type
Integer	long	TINYINT, SMALLINT, INT, and BIGINT
Floating point	double	FLOAT and DOUBLE
String	string	STRING, CHAR, VARCHAR, STRUCT, MAP, ARRAY, UNION, and BINARY
Date and time	date	DATE and TIMESTAMP
Boolean	boolean	BOOLEAN

Note

- LONG: data of the integer type in HDFS files, such as *123456789*.
- DOUBLE: data of the floating point type in HDFS files, such as *3.1415*.
- BOOLEAN: data of the Boolean type in HDFS files, such as *true* or *false*. Data is not case-sensitive.
- DATE: data of the date and time type in HDFS files, such as *2014-12-31 00:00:00*.

The TIMESTAMP data type supported by Hive can be accurate to the nanosecond. Therefore, data of the TIMESTAMP type stored in text and ORC files is similar to `2015-08-21 22:40:47.397898389`. After the data is converted to the DATE type in Data Integration, the nanosecond part in the data is lost. Therefore, you must specify the type of the converted data to STRING to make sure that the nanosecond part of the data is retained after conversion.

Parameters

Parameter	Description	Required	Default value
path	The path of the file from which you want to read data. If you want to read data from multiple files, you can specify a regular expression, such as `/hadoop/data_201704*`. If the file names contain time information and the time information is presented in a regular manner, you can use scheduling parameters together with a regular expression. The values of the scheduling parameters are replaced based on the data timestamp of the node. For more information about scheduling parameters, see Overview of scheduling parameters. • If you specify a single file, HDFS Reader uses only one thread to read data from the file. • If you specify multiple files, HDFS Reader uses multiple threads to read data from the files. The number of threads is determined by the concurrent parameter. Note The number of threads that are actually started is always the same as the smaller value between the number of HDFS files that you want to read and the number of parallel threads that you configure. • If a path contains a wildcard, HDFS Reader attempts to read data from all files that match the path. For example, if you specify the path as /bazhen/, HDFS Reader reads all files in the bazhen directory. HDFS Reader supports only asterisks `(*)` and question marks `(?)` as wildcards. The syntax is similar to the syntax of file name wildcards used in the Linux command line. Take note of the following items when you specify the path parameter: • Data Integration considers all the files to read in a synchronization node as a single table. Make sure that all the files can adapt to the same schema and Data Integration has the permissions to read all these files. • Table partitioning: When you create Hive tables, you can specify partitions. For example, if you specify `partition(day="20150820", hour="09")` when you create a Hive table, a directory named /20150820 and a subdirectory named /09 are created in the table directory in HDFS. Partitions form a directory structure. If you want to read all the data in a partition of a table, specify the path in the path parameter. For example, if you want to read all the data in the 20150820 partition in the table named mytable01, specify the path in the following way: `"path": "/user/hive/warehouse/mytable01/20150820/*"`	Yes	No default value
defaultFS	The endpoint of the NameNode node in HDFS. The shared resource group does not support advanced Hadoop parameters related to high availability.	Yes	No default value

Parameter	Description	Required	Default value
fileType	The format of the file from which you want to read data. HDFS Reader automatically identifies the file format and uses the related read policies. Before HDFS Reader reads data in a synchronization node, it checks whether all the files in the specified path match the format specified by the fileType parameter. If the format of a file does not match the format specified by the fileType parameter, the synchronization node fails. Valid values of the fileType parameter: - *TEXT*: the text format. - *ORC*: the ORC format. - *RC*: the RC format. - *SEQ*: the Sequence format. - *CSV*: the CSV format, which is a common HDFS file format. The data in a CSV file is organized as a logical two-dimensional table. - *PARQUET*: the Parquet format. HDFS Reader parses files in text and ORC formats in different ways. If data is converted from a Hive complex data type to the STRING type supported by Data Integration, the conversion results are different for the text and ORC formats. Complex data types include MAP, ARRAY, STRUCT, and UNION. The following examples demonstrate the results of the conversion from the MAP type to the STRING type: - After HDFS Reader parses and converts MAP-type data in an ORC file to the STRING type, the result is `{job=80, team=60, person=70}` . - After HDFS Reader parses and converts MAP-type data in a text file to the STRING type, the result is `{job:80, team:60, person:70}` . The conversion results show that the data remains unchanged but the formats differ slightly. Therefore, if a column that you want to synchronize uses a Hive complex data type, we recommend that you use a uniform file format. Recommended best practices: - To use a uniform file format, we recommend that you convert text files to ORC files on your Hive client. - If the file format is Parquet, you must specify the parquetSchema parameter, which specifies the schema of data in Parquet files.	Yes	No default value

Parameter	Description	Required	Default value

column	The names of the columns from which you want to read data. The type field specifies a data type. The index field specifies the ID of a column, starting from 0. The value field specifies a constant. If you specify the value field, HDFS Reader reads the value of this field. By default, HDFS Reader reads all data as strings. In this case, set this parameter to `"column": ["*"]` .
	For the column parameter, you must configure the type parameter and one of the index and value parameters. Example:
	`{`
	`  "type": "long",`
	`  "index": 0`
	`  // The first INT-type column of the source file. The index starts from 0. The index field indicates the IDs of the columns from which you want to read data in the file.`
	`},`
	`{`
	`  "type": "string",`
	`  "value": "alibaba"`
	`  // The value of the current column, which is a constant column alibaba. It is internally generated by HDFS Reader.`
	`}`
	Note
	- The index starts from 0, which indicates that HDFS Reader reads data from the first column of the source file.
	- We recommend that you specify the index and type fields for each column from which you want to read data, instead of using `column *` .
		Yes	No default value

fieldDelimiter	The delimiter of the columns from which you want to read data. If the source files are text files, you must specify a column delimiter. If you do not specify a column delimiter, HDFS Reader uses commas (,) as column delimiters by default. If the source files are ORC files, you do not need to specify a column delimiter. HDFS Reader uses the default delimiter of Hive, which is \u0001. Note - If you want each row to be converted to a column in the destination file, use a character that does not exist in these rows as the delimiter, such as \u0001. - Do not use \n as the delimiter.	No	,
encoding	The encoding format of the file from which you want to read data.	No	utf-8
nullFormat	The string that represents a null pointer. No standard strings can represent a null pointer in TXT files. You can use this parameter to define which string represents a null pointer. For example, if you set this parameter to `null` , Data Integration considers null as a null pointer. Note The string NULL is different from a null pointer. Pay attention to the difference between them.	No	No default value

Parameter	Description	Required	Default value
compress	The compression format when the fileType parameter is set to CSV. The following compression formats are supported: GZIP, BZ2, ZIP, LZO, LZO_DEFLATE, Hadoop-Snappy, and Framing-Snappy. Note - LZO and LZO_DEFLATE are two different compression formats. Do not mix them up when you configure this parameter. - Snappy does not have a uniform stream format. Data Integration supports only the two most popular compression formats: Hadoop-Snappy and Framing-Snappy. Hadoop-Snappy is the Snappy stream format in Hadoop, and Framing-Snappy is the Snappy stream format recommended by Google. - This parameter is not required if you set the fileType parameter to ORC.	No	No default value

parquetSchema	The description of the schema of data in Parquet files. If you set the fileType parameter to Parquet, you must set the parquetSchema parameter. Make sure that the value of the parquetSchema parameter complies with the JSON syntax.
	`message MessageTypeName {`
	`required, dataType, columnName;`
	`......................;`
	`}`
	The parquetSchema parameter contains the following fields:
	- MessageTypeName: the name of the MessageType object.
	- required: indicates that the field cannot be empty. The value optional indicates that the field can be empty. We recommend that you set this parameter to optional for all fields.
	- dataType: Parquet files support various field types such as BOOLEAN, INT32, INT64, INT96, FLOAT, DOUBLE, BINARY, and FIXED_LEN_BYTE_ARRAY. Set this parameter to BINARY if the field stores strings.
	- Each line, including the last one, must end with a semicolon (;).
	Configuration example:
	`"parquetSchema": "message m { optional int32 minute_id; optional int32 dsp_id; optional int32 adx_pid; optional int64 req; optional int64 res; optional int64 suc; optional int64 imp; optional double revenue; }"`
		No	No default value

Parameter	Description	Required	Default value

csvReaderConfig	The configurations required to read CSV files. The parameter value must match the MAP type. You can use a CSV file reader to read data from CSV files. The CSV file reader supports multiple configurations. If you do not configure this parameter, the default configurations are used.
	The following example shows common configurations:
	`"csvReaderConfig":{`
	`  "safetySwitch": false,`
	`  "skipEmptyRecords": false,`
	`  "useTextQualifier": false`
	`}`
	The following configurations show all the fields and their default values. When you configure the csvReaderConfig parameter of the MAP type, you must use the field names provided in the following configurations:
	`boolean caseSensitive = true;`
	`char textQualifier = 34;`
	`boolean trimWhitespace = true;`
	`boolean useTextQualifier = true;// Specifies whether to use escape characters for CSV files.`
	`char delimiter = 44;// The delimiter.`
	`char recordDelimiter = 0;`
	`char comment = 35;`
	`boolean useComments = false;`
	`int escapeMode = 1;`
	`boolean safetySwitch = true;// Specifies whether to limit the length of each column to 100,000 characters.`
	`boolean skipEmptyRecords = true;// Specifies whether to skip empty rows.`
	`boolean captureRawRecord = true;`
		No	No default value

Parameter	Description	Required	Default value

hadoopConfig	The settings of advanced Hadoop parameters, such as the parameters related to high availability. The shared resource group does not support advanced Hadoop parameters related to high availability.
	`"hadoopConfig":{`
	`"dfs.nameservices": "testDfs",`
	`"dfs.ha.namenodes.testDfs": "namenode1,namenode2",`
	`"dfs.namenode.rpc-address.youkuDfs.namenode1": "",`
	`"dfs.namenode.rpc-address.youkuDfs.namenode2": "",`
	`"dfs.client.failover.proxy.provider.testDfs":`
	`"org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider",`
	`"dfs.data.transfer.protection": "integrity",`
	`"dfs.datanode.use.datanode.hostname" :"true",`
	`"dfs.client.use.datanode.hostname":"true"`
	`}`
	Note
	`"hadoopConfig":{ "dfs.data.transfer.protection": "integrity", "dfs.datanode.use.datanode.hostname" :"true", "dfs.client.use.datanode.hostname":"true" }`
	The preceding settings are used to configure Kerberos authentication in HDFS Reader. If you configure Kerberos authentication in the HDFS data source, you do not need to configure it in HDFS Reader. For more information about how to add an HDFS data source, see Add an HDFS data source.
		No default value	

haveKerberos	Specifies whether Kerberos authentication is required. Default value: false. If you set this parameter to true, you must also configure the kerberosKeytabFilePath and kerberosPrincipal parameters.	No	false
kerberosKeytabFilePath	The absolute path of the keytab file for Kerberos authentication. This parameter is required if the haveKerberos parameter is set to true.	No	No default value

kerberosPrincipal	The Kerberos principal to which Kerberos can assign tickets, such as ****/hadoopclient@**.***. This parameter is required if the haveKerberos parameter is set to true.
	Note The absolute path of the keytab file is required for Kerberos authentication. Therefore, you must configure Kerberos authentication for exclusive resource groups for Data Integration. The following code provides a configuration example:
	`"haveKerberos":true,`
	`"kerberosKeytabFilePath":"/opt/datax/**.keytab",`
	`"kerberosPrincipal":"**/hadoopclient@**.**"`
		No	No default value

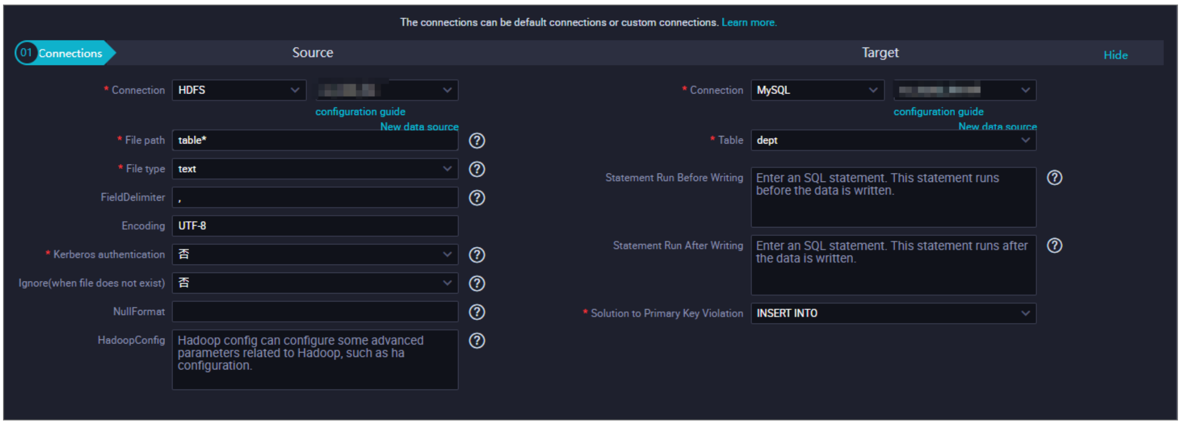
Configure HDFS Reader by using the codeless UI

Create a synchronization node and configure the node. For more information, see Configure a synchronization node by using the codeless UI.

Perform the following steps on the configuration tab of the synchronization node:

1. Configure data sources.

Configure **Source** and **Target** for the synchronization node.



The connections can be default connections or custom connections. Learn more.

01 Connections Source Target Hide

Source:

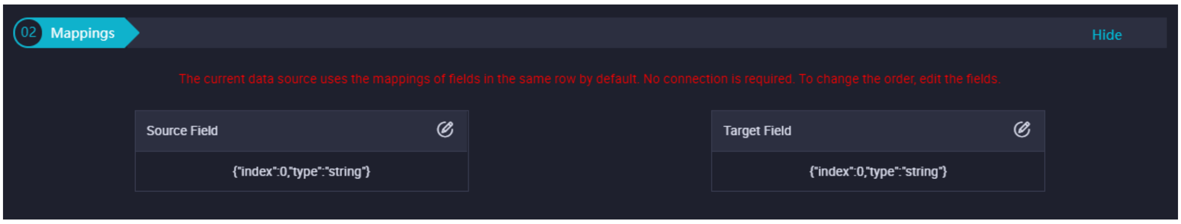
- * Connection: HDFS [redacted] configuration guide New data source
- * File path: table*
- * File type: text
- FieldDelimiter: ,
- Encoding: UTF-8
- * Kerberos authentication: 否
- Ignore(when file does not exist): 否
- NullFormat:
- HadoopConfig: Hadoop config can configure some advanced parameters related to Hadoop, such as ha configuration.

Target:

- * Connection: MySQL [redacted] configuration guide New data source
- * Table: dept
- Statement Run Before Writing: Enter an SQL statement. This statement runs before the data is written.
- Statement Run After Writing: Enter an SQL statement. This statement runs after the data is written.
- * Solution to Primary Key Violation: INSERT INTO

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
File path	The path of the file from which you want to read data. This parameter is equivalent to the path parameter that is described in the preceding section.
File type	This parameter is equivalent to the fileType parameter that is described in the preceding section. This parameter specifies the format of the file from which you want to read data. Valid values: *text*, *orc*, *rc*, *seq*, *csv*, and *parquet*.
FieldDelimiter	The column delimiter. This parameter is equivalent to the fieldDelimiter parameter that is described in the preceding section. By default, a comma (,) is used as a column delimiter.
Encoding	The encoding format. This parameter is equivalent to the encoding parameter that is described in the preceding section. Default value: UTF-8.
Kerberos authentication	Specifies whether to enable Kerberos authentication. Default value: No. If you set this parameter to Yes, the KeyTab file path and Principal Name parameters are required. For more information, see Configure Kerberos authentication.
Ignore(when file does not exist)	Specifies whether to ignore the file or folder that you configure if the file or folder does not exist. If you set this parameter to Yes, HDFS Reader does not read data from the file, and no error is reported in the log. If you set this parameter to No, the data synchronization node fails to run. Default value: No.
NullFormat	This parameter is equivalent to the NullFormat parameter that is described in the preceding section. This parameter specifies the string that represents a null pointer.
HadoopConfig	The settings of advanced Hadoop parameters, such as the parameters related to high availability. The shared resource group does not support advanced Hadoop parameters related to high availability.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. The system maps the field in a row of the source to the field in the same row of the destination. You can click the ✎ icon to manually edit the fields in the source. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.



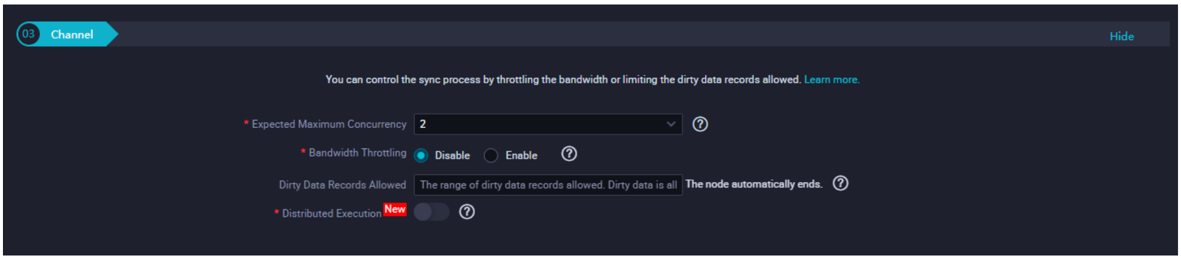
02 Mappings Hide

The current data source uses the mappings of fields in the same row by default. No connection is required. To change the order, edit the fields.

Source Field	Target Field
{"index":0,"type":"string"}	{"index":0,"type":"string"}

> Note The index starts from 0, which indicates that HDFS Reader reads data from the first column of the source file.

3. Configure channel control policies.



03 Channel Hide

You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

- * Expected Maximum Concurrency: 2
- * Bandwidth Throttling: ● Disable ○ Enable
- Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.
- * Distributed Execution New

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure HDFS Reader by using the code editor

In the following code, a synchronization node is configured to read data from HDFS. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

> Note You must delete the comments from the following code before you run the code.

```
{
    "type": "job",
    "version": "2.0",
    "steps": [
        {
            "stepType": "hdfs",// The reader type.
            "parameter": {
                "path": "",// The path of the file from which you want to read data.
                "datasource": "",// The name of the data source.
                "hadoopConfig":{
                "dfs.data.transfer.protection": "integrity",
               "dfs.datanode.use.datanode.hostname" :"true",
                "dfs.client.use.datanode.hostname":"true"
                 },
                "column": [
                    {
                        "index": 0,// The index of the column in the source file. The index starts from 0, whic
h indicates that HDFS Reader reads data from the first column of the source file.
```

```
                        "type": "string"// The field type.
                    },
                    {
                        "index": 1,
                        "type": "long"
                    },
                    {
                        "index": 2,
                        "type": "double"
                    },
                    {
                        "index": 3,
                        "type": "boolean"
                    },
                    {
                        "format": "yyyy-MM-dd HH:mm:ss",// The time format.
                        "index": 4,
                        "type": "date"
                    }
                ],
                "fieldDelimiter": ","// The column delimiter.
                "encoding": "UTF-8",// The encoding format.
                "fileType": ""// The file format.
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "stream",
            "parameter": {},
            "name": "Writer",
            "category": "writer"
        }
    ],
    "setting": {
        "errorLimit": {
            "record": ""// The maximum number of dirty data records allowed.
        },
        "speed": {
            "concurrent": 3,// The maximum number of parallel threads.
            "throttle": true // Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}
```

The following example shows the HDFS Reader configuration with the parquetSchema parameter.

Note

- The fileType parameter must be set to PARQUET.
- If you want HDFS Reader to read specific columns from a Parquet file, you must specify the complete schema in the parquetSchema parameter and specify the columns that you want to read by using the index field in the column parameter.

```
"reader":  {
    "name": "hdfsreader",
    "parameter": {
        "path": "/user/hive/warehouse/addata.db/dw_ads_rtb_monitor_minute/thedate=20170103/hour_id=22/*",
        "defaultFS": "h10s010.07100.149:8020",
        "column": [
            {
                "index": 0,
                "type": "string"
            },
            {
                "index": 1,
                "type": "long"
            },
            {
                "index": 2,
                "type": "double"
            }
        ],
        "fileType": "parquet",
        "encoding": "UTF-8",
        "parquetSchema": "message m { optional int32 minute_id; optional int32 dsp_id; optional int32 adx_pid;
optional int64 req; optional int64 res; optional int64 suc; optional int64 imp; optional double revenue; }"
    }
}
```

5.2.5. MongoDB Reader

This topic describes the data types and parameters that are supported by MongoDB Reader and how to configure MongoDB Reader by using the codeless user interface (UI) and code editor.

MongoDB Reader connects to a remote MongoDB database by using the Java client MongoClient and reads data from the database. The locking feature in the latest version of MongoDB is improved from database-level locking to document-level locking. This enables MongoDB Reader to efficiently read data from MongoDB databases by using the powerful indexing capabilities in MongoDB.

Note

- If you use ApsaraDB for MongoDB, the MongoDB database has a root account by default. For security purposes, Data Integration can access a MongoDB database only by using a MongoDB database account. When you add a MongoDB data source, do not use the root account for access.
- The query parameter does not support the JavaScript syntax.

MongoDB Reader shards data in a MongoDB database based on specific rules, reads data from the database by using parallel threads, and then converts the data to a format that is readable to Data Integration.

Data types

MongoDB Reader supports most MongoDB data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by MongoDB Reader.

Data Integration data type	MongoDB data type

Data Integration data type	MongoDB data type
LONG	INT, LONG, document.INT, and document.LONG
DOUBLE	DOUBLE and document.DOUBLE
STRING	STRING, ARRAY, document.STRING, document.ARRAY, and COMBINE
DATE	DATE and document.DATE
BOOLEAN	BOOLEAN and document.BOOLEAN
BYTES	BYTES and document.BYTES

Note The DOCUMENT data type is used to store embedded documents. It is also called the OBJECT data type.

When you use the COMBINE data type, take note of the following items:

When MongoDB Reader reads data from a MongoDB database, MongoDB Reader combines multiple fields in MongoDB documents into a JSON string.

For example, doc1, doc2, and doc3 are three MongoDB documents that contain different fields. The fields are represented by keys instead of key-value pairs. The keys a and b are common fields in all of the three documents. The key x_n represents a document-specific field.

```
doc1: a b x_1 x_2
```

```
doc2: a b x_2 x_3 x_4
```

```
doc3: a b x_5
```

To import the preceding three MongoDB documents to MaxCompute, you must specify the fields that you want to retain, specify a name for each JSON string that is obtained, and specify the data type of each obtained JSON string to COMBINE in the configuration file. Make sure that the name of each obtained JSON string is different from that of an existing field in the documents.

```
"column": [
{
"name": "a",
"type": "string",
},
{
"name": "b",
"type": "string",
},
{
"name": "doc",
"type": "combine",
}
]
```

The following table lists the output in MaxCompute.

odps_column1	odps_column2	odps_column3
a	b	{x_1,x_2}
a	b	{x_2,x_3,x_4}
a	b	{x_5}

Note

When you combine multiple fields in a MongoDB document and set the data type of each obtained JSON string to COMBINE, the result that is exported to MaxCompute contains only fields specific to the document. Common fields are automatically deleted.

In the preceding example, a and b are common fields in all of the three documents. After fields in the document file `doc1: a b x_1 x_2` are combined and the data type of the obtained JSON strings is set to COMBINE, the result is *{a,b,x_1,x_2}*. When the result is exported to MaxCompute, common fields a and b are deleted, and the result is *{x_1,x_2}*.

Limits

- A maximum of one parallel thread can be used to read data from the source or write data to the destination in a synchronization node that uses MongoDB Reader.
- The shard key must be a field of an integer data type. Otherwise, non-consecutive shards may be generated, and data may be lost.

Parameters

Parameter	Description
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.
collectionName	The name of the collection in the MongoDB database.
column	The names of the document fields from which you want to read data. Specify the names in an array. - *name*: the name of a field. - *type*: the data type of a field. Valid values: ○ *string*: string. ○ *long*: integer. ○ *double*: floating point. ○ *date*: date. ○ *bool*: Boolean. ○ *bytes*: binary. ○ *arrays*: MongoDB Reader reads data from the MongoDB documents as a JSON array, such as ["a","b","c"]. ○ *array*: MongoDB Reader reads data from the MongoDB documents as a common array, in which elements are separated by delimiters, such as `a,b,c`. We recommend that you set type to *arrays*. ○ *combine*: MongoDB Reader combines multiple fields in the MongoDB documents into a JSON string. - *splitter*: the delimiter. Configure this parameter only if you want to convert an array to a string. MongoDB supports arrays, but Data Integration does not. The array elements that are read by MongoDB Reader are joined into a string by using this delimiter.
batchSize	The number of data records that are read at a time. This parameter is optional. Default value: `1000`.

Parameter	Description
cursorTimeoutInMs	The timeout period of the cursor. Unit: milliseconds. This parameter is optional. Default value: `600000`. The default value 600000 is equivalent to 10 minutes. If you set this parameter to a negative number, the cursor never times out. Note - We recommend that you do not set this parameter to a negative number. If you set this parameter to a negative number and the MongoDB client unexpectedly exits, the cursor that never times out persists in the MongoDB server until the MongoDB client is restarted. - If the cursor times out, you can perform one of the following operations to fix the issue: - Specify a small value for the *batchSize* parameter. - Specify a large value for the *cursorTimeoutInMs* parameter.
query	The condition that is used to filter data from MongoDB. Only data of the time type is supported. For example, you can specify `"query":"{'operationTime':{'$gte':ISODate('${last_day}T00:00:00.424+0800')}}"` to obtain data in which the time that is specified by operationTime is not earlier than 00:00 on the day that is specified by ${last_day}. ${last_day} is a scheduling parameter of DataWorks. Specify *last_day* in the `yyyy-mm-dd` format. You can use comparison operators such as $gt, $lt, $gte, and $lte, logical operators such as "and" and "or", and functions such as max, min, sum, avg, and ISODate that are supported by MongoDB based on your business requirements. This parameter is optional.

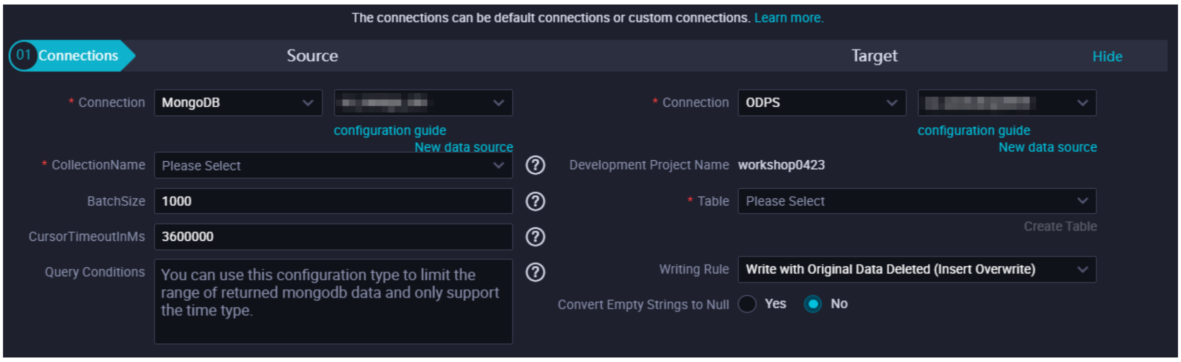
Configure MongoDB Reader by using the codeless UI

Create a synchronization node and configure the node. For more information, see Configure a synchronization node by using the codeless UI.

You must perform the following steps on the configuration tab of the synchronization node:

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



The connections can be default connections or custom connections. Learn more.

01 Connections

Source

* Connection: MongoDB, [redacted]

configuration guide

New data source

* CollectionName: Please Select

BatchSize: 1000

CursorTimeoutInMs: 3600000

Query Conditions: You can use this configuration type to limit the range of returned mongodb data and only support the time type.

Target

Hide

* Connection: ODPS, [redacted]

configuration guide

New data source

Development Project Name: workshop0423

* Table: Please Select

Create Table

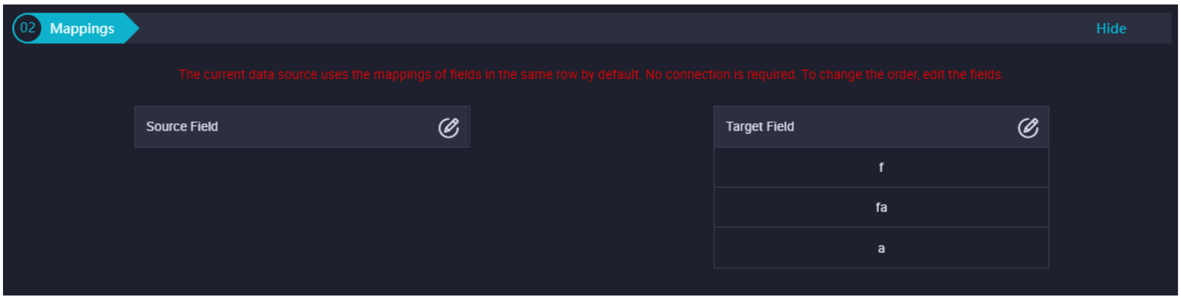
Writing Rule: Write with Original Data Deleted (Insert Overwrite)

Convert Empty Strings to Null: ○ Yes ● No

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
CollectionName	The name of the collection in the MongoDB database. This parameter is equivalent to the collectionName parameter that is described in the preceding section.
BatchSize	The number of data records to read from the MongoDB database at a time. Default value: 1000.
CursorTimeoutInMs	The timeout period of the cursor. Default value: 3600000. Unit: milliseconds. If you set this parameter to a negative number, the cursor never times out.

Parameter	Description
Query Conditions	This parameter is equivalent to the query parameter that is described in the preceding section. You can configure this parameter to filter data from MongoDB.

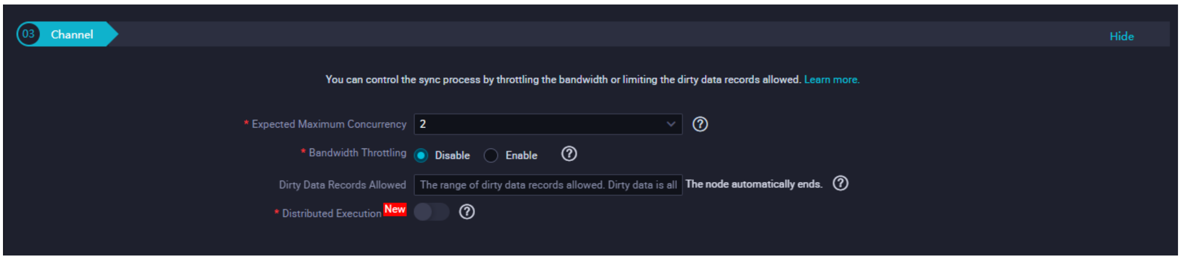
2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. By default, the system maps the field in a row of the source to the field in the same row of the destination. You can click the icon to manually edit fields in the MongoDB documents.



The current data source uses the mappings of fields in the same row by default. No connection is required. To change the order, edit the fields.

Source Field	Target Field
	f
	fa
	a

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node can use to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI. **Note** You can set this parameter only to 1.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure MongoDB Reader by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to read data from a MongoDB database. For more information about the parameters, see the preceding parameter description.

Notice

- Delete the comments from the following code before you run the code.
- MongoDB Reader cannot read some elements in arrays.

```
{
    "type":"job",
    "version":"2.0", // The version number.
    "steps":[
        {
            "category": "reader",
            "name": "Reader",
            "parameter": {
                "datasource": "datasourceName", // The name of the data source.
                "collectionName": "tag_data", // The name of the collection in the MongoDB database.
                "query": "", // The condition that is used to filter data from MongoDB.
                "column": [
                    {
                        "name": "unique_id", // The name of the field.
                        "type": "string" // The data type of the field.
                    },
                    {
                        "name": "sid",
                        "type": "string"
                    },
                    {
                        "name": "user_id",
                        "type": "string"
                    },
                    {
                        "name": "auction_id",
                        "type": "string"
                    },
                    {
                        "name": "content_type",
                        "type": "string"
                    },
                    {
                        "name": "pool_type",
                        "type": "string"
                    },
                    {
                        "name": "frontcat_id",
                        "type": "array",
                        "splitter": ""
                    },
                    {
                        "name": "categoryid",
                        "type": "array",
                        "splitter": ""
                    },
                    {
                        "name": "gmt_create",
                        "type": "string"
                    },
                    {
                        "name": "taglist",
                        "type": "array",
                        "splitter": " "
                    },
                    {
                        "name": "property",
                        "type": "string"
                    },
                    {
                        "name": "scorea",
                        "type": "int"
                    },
                    {
                        "name": "scoreb",
                        "type": "int"
                    },
                    {
                        "name": "scorec",
                        "type": "int"
                    },
                    {
                        "name": "a.b",
                        "type": "document.int"
                    },
                    {
                        "name": "a.b.c",
                        "type": "document.array",
                        "splitter": " "
                    }
                ]
            },
            "stepType": "mongodb"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1 // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.2.6. DB2 Reader

This topic describes the data types and parameters that are supported by Db2 Reader and how to configure Db2 Reader by using the codeless user interface (UI) and code editor.

Notice Db2 Reader supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration, Use the default resource group, and Create a custom resource group for Data Integration.

Background information

Db2 Reader can read data from Db2.

Db2 Reader connects to a remote Db2 database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the statement on the database and returns data. Then, Db2 Reader assembles the returned data into abstract datasets of the data types supported by **Data Integration** and sends the datasets to a writer.

- Db2 Reader generates an SQL statement based on the settings of the table, column, and where parameters and sends the generated statement to the Db2 database.
- If you specify the querySql parameter, Db2 Reader sends the value of this parameter to the Db2 database.

The version of the Db2 JDBC driver that Db2 Reader uses is IBM Data Server Driver for JDBC and SQLJ 4.11.77. For more information about the mapping between the versions of Db2 JDBC drivers and Db2 versions, see IBM Support.

Data types

Db2 Reader supports most Db2 data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by Db2 Reader.

Data Integration data type	Db2 data type
Integer	SMALLINT
Floating point	DECIMAL, REAL, and DOUBLE
String	CHAR, CHARACTER, VARCHAR, GRAPHIC, VARGRAPHIC, LONG VARCHAR, CLOB, LONG VARGRAPHIC, and DBCLOB
Date and time	DATE, TIME, and TIMESTAMP
Binary	BLOB

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
jdbcUrl	The JDBC URL of the Db2 database. The URL must be in the `jdbc:db2://ip:port/database` format in accordance with official Db2 specifications. You can also specify the information of the attachment facility.	Yes	No default value
username	The username that you use to connect to the database.	Yes	No default value
password	The password that you use to connect to the database.	Yes	No default value
table	The name of the table from which you want to read data. Each synchronization node can be used to synchronize data to only one table.	Yes	No default value

Parameter	Description	Required	Default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. The default value is [*], which indicates all the columns in the source table. • You can select specific columns to read. • The column order can be changed. This indicates that you can specify columns in an order different from the order specified by the schema of the source table. • Constants are supported. The column names must be arranged in compliance with the SQL syntax that is supported by Db2, such as `["id", "1", "'const name'", "null", "upper('abc_lower')", "2.3" , "true"]` . ○ id: a column name. ○ 1: an integer constant. ○ 'const name': a string constant, which is enclosed in single quotation marks ('). ○ null: a null pointer. ○ upper('abc_lower'): a function expression. ○ 2.3: a floating-point constant. ○ true: a Boolean value. • The column parameter must explicitly specify all the columns from which you want to read data. This parameter cannot be left empty.	Yes	No default value
splitPk	The field that is used for data sharding when Db2 Reader reads data. If you specify this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This improves data synchronization efficiency. • We recommend that you set the splitPk parameter to the name of the primary key column of the table. Data can be evenly distributed to different shards based on the primary key column, but not intensively distributed only to specific shards. • The splitPk parameter supports sharding only for data of integer data types. If you set this parameter to a column of an unsupported data type, such as a string, floating point, or date data type, an error is reported.	No	""
where	The WHERE clause. Db2 Reader generates an SQL statement based on the settings of the table, column, and where parameters and uses the generated statement to read data. For example, you can set this parameter to `gmt_create > $bizdate` in an actual business scenario to read data that is generated on the current day. You can use the WHERE clause to read incremental data. If this parameter is left empty, Db2 Reader reads all the data in the source table.	No	No default value
querySql	The SQL statement that is used for refined data filtering. If you specify this parameter, Data Integration filters data based on the value of this parameter. For example, if you want to join multiple tables for data synchronization, set this parameter to `select a,b from table_a join table_b on table_a.id = table_b.id` . If you specify this parameter, Db2 Reader ignores the settings of the table, column, and where parameters.	No	No default value
fetchSize	The number of data records to read at a time. This parameter determines the number of interactions between Data Integration and the database and affects read efficiency. Note If you set this parameter to a value greater than 2048, an out of memory (OOM) error may occur during data synchronization.	No	*1024*

Configure Db2 Reader by using the codeless UI

This method is not supported.

Configure Db2 Reader by using the code editor

In the following code, a synchronization node is configured to read data from a Db2 database. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"db2",// The reader type.
            "parameter":{
                "password":"",// The password that you use to connect to the Db2 database.
                "jdbcUrl":"",// The JDBC URL of the Db2 database.
                "column":[
                    "id"
                ],
                "where":"",// The WHERE clause.
                "splitPk":"",// The shard key.
                "table":"",// The name of the table from which you want to read data.
                "username":""// The username that you use to connect to the Db2 database.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
             "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates tha
t bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps
parameter takes effect only when the throttle parameter is set to true.
             "concurrent":1 // The maximum number of parallel threads.
             "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

Additional information

- Data synchronization between primary and secondary databases

  A secondary Db2 database can be deployed for disaster recovery. The secondary database continuously synchronizes data from the primary database based on binary logs. Data latency between the primary and secondary databases cannot be prevented. This may result in data inconsistency.

- Data consistency control

Db2 is a relational database management system (RDBMS) that supports strong consistency for data queries. A database snapshot is created before a synchronization node starts. Db2 Reader reads data from the database snapshot. Therefore, if new data is written to the database during data synchronization, Db2 Reader cannot obtain the new data.

Data consistency cannot be ensured if you enable Db2 Reader to use parallel threads in a single synchronization node.

Db2 Reader shards the source table based on the value of the splitPk parameter and uses parallel threads to read data. These parallel threads belong to different transactions and read data at different points in time. Therefore, the parallel threads observe different snapshots.

Data inconsistencies cannot be prevented if parallel threads are used for a synchronization node. The following workarounds can be used:

  - Enable Db2 Reader to use a single thread to read data in a synchronization node. This indicates that you do not need to specify a shard key for Db2 Reader. This way, data consistency is ensured, but data is synchronized at low efficiency.
  - Make sure that no data is written to the source table during data synchronization. This ensures that the data in the source table remains unchanged during data synchronization. For example, you can lock the source table or disable data synchronization between primary and secondary databases. This way, data is efficiently synchronized, but your ongoing services may be interrupted.

- Character encoding

  Db2 Reader uses JDBC to read data. This enables Db2 Reader to automatically convert the encoding formats of characters. Therefore, you do not need to specify the encoding format.

- Incremental data synchronization

  Db2 Reader uses JDBC to connect to a database and uses a SELECT statement with a `WHERE` clause to read incremental data.

  - For batch data, incremental add, update, and delete operations (including logically delete operations) are distinguished by timestamps. Specify the WHERE clause based on a specific timestamp. The time indicated by the timestamp must be later than the time indicated by the latest timestamp in the previous synchronization.
  - For streaming data, specify the WHERE clause based on the ID of a specific record. The ID must be greater than the maximum ID involved in the previous synchronization.

  If the data that is added or modified cannot be distinguished, Db2 Reader can read only full data.

- Syntax validation

  Db2 Reader allows you to specify custom SELECT statements by using the querySql parameter but does not verify the syntax of these statements.

5.2.7. MySQL Reader

This topic describes the data types and parameters that are supported by MySQL Reader and how to configure MySQL Reader by using the codeless user interface (UI) and code editor.

Prerequisites

A MySQL data source is configured. For more information, see Add a MySQL data source.

Background information

MySQL Reader connects to a remote MySQL database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the statement on the database and returns data. Then, MySQL Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

MySQL Reader can read data from tables and views. If you want to read data from a table, you can specify all or some of the columns in the table in an order that is specified by the table schema, specify constant fields, or configure MySQL functions such as now(). You can also specify the columns in an order that is different from the order specified by the table schema.

Data types

The following table lists the data types that are supported by MySQL Reader.

Category	MySQL data type
Integer	INT, TINYINT, SMALLINT, MEDIUMINT, and BIGINT
Floating point	FLOAT, DOUBLE, and DECIMAL
String	VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT
Date and time	DATE, DATETIME, TIMESTAMP, TIME, and YEAR
Boolean	BIT and BOOLEAN
Binary	TINYBLOB, MEDIUMBLOB, BLOB, LONGBLOB, and VARBINARY

Notice

- Data types that are not listed in the preceding table are not supported.
- MySQL Reader processes TINYINT(1) as an integer data type.

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table from which you want to read data. Each synchronization node can be used to synchronize data to only one table. For a sharded table, you can use the table parameter to specify the partitions from which you want to read data. Examples: • Set the table parameter to `'table_[0-99]'`. This value indicates that MySQL Reader reads data from the partitions `'table_0'` to `'table_99'` of the sharded table. • Set the table parameter to `'"table":["table_00[0-9]", "table_0[10-99]", "table_[100-999]"]'`. This value indicates that MySQL Reader reads data from the partitions `'table_000'` to `'table_999'` of the sharded table. You can use this method only if the numerical suffixes of all the partition names are of the same length. Note MySQL Reader reads data from the columns that are specified by the column parameter in the partitions that are specified by the table parameter. If a specified partition or column does not exist, the synchronization node fails.	Yes	No default value

Parameter	Description	Required	Default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. The default value is [*], which indicates all the columns in the source table. • You can select specific columns to read. • The column order can be changed. This indicates that you can specify columns in an order different from the order specified by the schema of the source table. • Constants are supported. The column names must be arranged in compliance with the SQL syntax supported by MySQL, such as `["id","table","1","'mingya.wmy'","'null'","to_char(a+1)","2.3","true"]` . ○ id: a column name. ○ table: the name of a column that contains reserved keywords. ○ 1: an integer constant. ○ 'mingya.wmy': a string constant, which is enclosed in single quotation marks ('). ○ null: ■ `" "` indicates an empty string. ■ `null` indicates a null value. ■ `'null'` indicates the string null. ○ to_char(a+1): a function expression that is used to calculate the length of a string. ○ 2.3: a floating-point constant. ○ true: a Boolean value. • The column parameter must explicitly specify all the columns from which you want to read data. The parameter cannot be left empty.	Yes	No default value
splitPk	The field that is used for data sharding when MySQL Reader reads data. If you specify this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This way, data can be synchronized more efficiently. • We recommend that you set the splitPk parameter to the name of the primary key column of the table. This way, data can be evenly distributed to different shards based on the primary key column, instead of being intensively distributed only to specific shards. • The splitPk parameter supports sharding only for data of integer data types. If you set the splitPk parameter to a field of an unsupported data type, such as a string, floating point, or date data type, the setting of this parameter is ignored, and a single thread is used to read data. • If the splitPk parameter is not provided or is left empty, a single thread is used to read data.	No	No default value
where	The WHERE clause. For example, you can set this parameter to `gmt_create > $bizdate` to read the data that is generated on the current day. • You can use the WHERE clause to read incremental data. If the where parameter is not provided or is left empty, MySQL Reader reads all data. • Do not set the where parameter to limit 10. This value does not conform to the constraints of MySQL on the SQL WHERE clause.	No	No default value

Parameter	Description	Required	Default value
querySql (advanced parameter, which is available only in the code editor)	The SQL statement that is used for refined data filtering. If you specify this parameter, data is filtered based only on the value of this parameter. For example, if you want to join multiple tables for data synchronization, set this parameter to `select a,b from table_a join table_b on table_a.id = table_b.id`. The priority of the querySql parameter is higher than the priorities of the table, column, where, and splitPk parameters. If you specify the querySql parameter, MySQL Reader ignores the settings of the table, column, where, and splitPk parameters. The system parses the information, such as the username and password, of the data source specified by the datasource parameter from the querySql parameter. **Note** The name of the querySql parameter is case-sensitive. For example, querysql does not take effect.	No	No default value
singleOrMulti (available only for sharded tables)	Specifies that the source table is a sharded table. After you switch from the codeless UI to the code editor, the `"singleOrMulti":"multi"` configuration is automatically generated. However, if you use the code editor at the beginning, the configuration is not automatically generated, and you must manually add the configuration in the code editor. If you do not add the configuration, MySQL Reader reads data only from the first shard.	Yes	*multi*

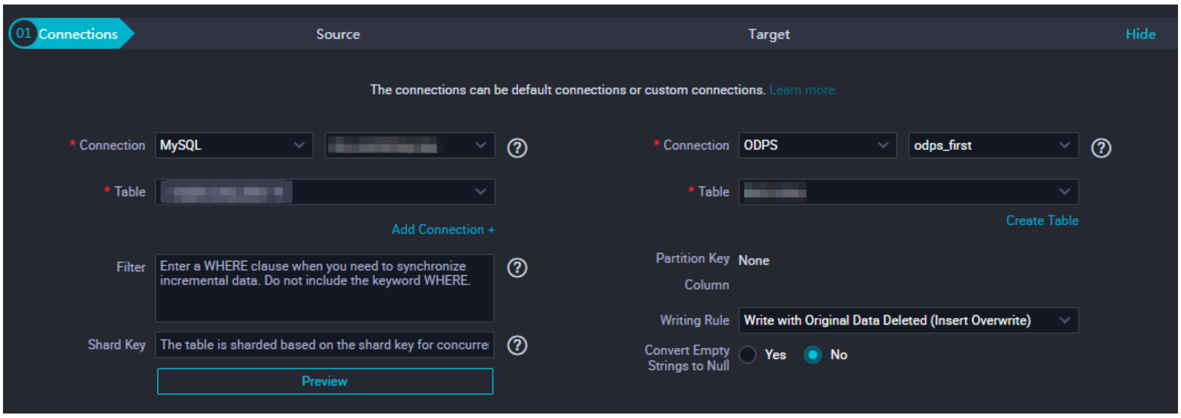
Configure MySQL Reader by using the codeless UI

Create a synchronization node and configure the node. For more information, see Configure a synchronization node by using the codeless UI.

Perform the following steps on the configuration tab of the synchronization node:

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



01 Connections

Source

Target

The connections can be default connections or custom connections. Learn more.

Source:
- * Connection: MySQL, [redacted]
- * Table: [redacted]
- Filter: Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.
- Shard Key: The table is sharded based on the shard key for concurre

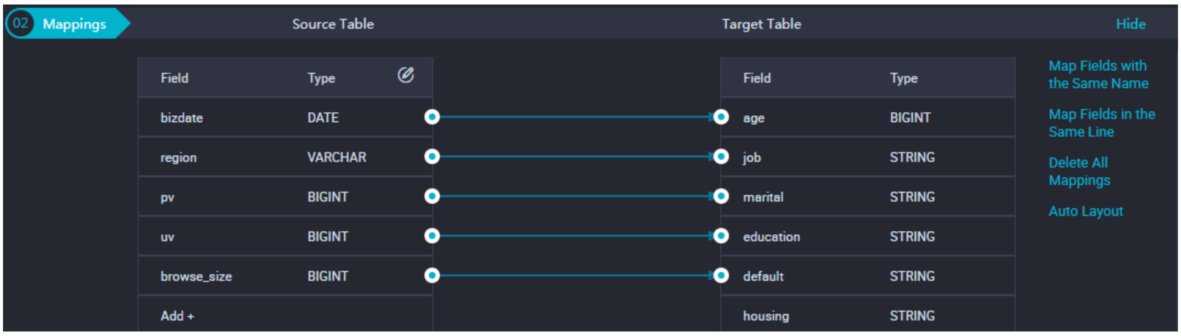
Target:
- * Connection: ODPS, odps_first
- * Table: [redacted]
- Partition Key Column: None
- Writing Rule: Write with Original Data Deleted (Insert Overwrite)
- Convert Empty Strings to Null: ○ Yes ● No

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table from which you want to read data. This parameter is equivalent to the table parameter that is described in the preceding section.
Filter	The condition that is used to filter the data you want to read. Filtering based on the LIMIT keyword is not supported. The SQL syntax is determined by the selected data source.

Parameter	Description
Shard Key	The shard key. You can use a column in the source table as the shard key. We recommend that you use the primary key column or an indexed column. Only integer columns are supported. If you specify this parameter, data sharding is performed based on the value of this parameter, and parallel threads can be used to read data. This improves data synchronization efficiency. ⓘ **Note** The Shard Key parameter is displayed only after you select the data source for the synchronization node.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



02 Mappings Source Table Target Table Hide

Field	Type	Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
Add +		housing	STRING

Map Fields with the Same Name

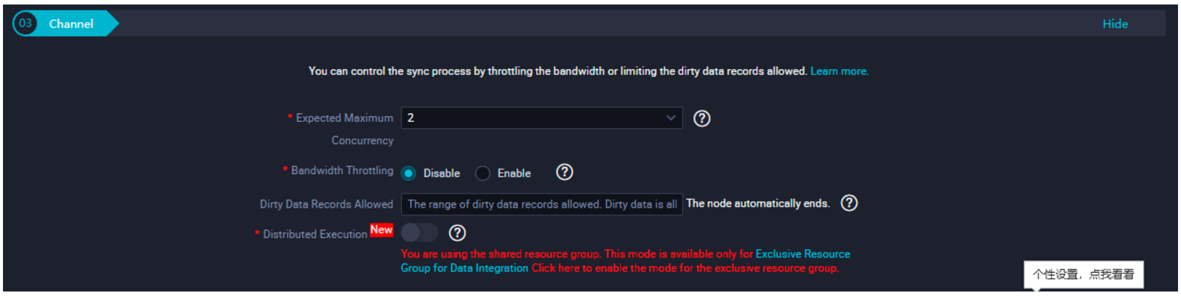
Map Fields in the Same Line

Delete All Mappings

Auto Layout

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	Click **Add** to add a field. Take note of the following rules when you add a field: ◦ You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ◦ You can use scheduling parameters, such as ${bizdate}. ◦ You can enter functions that are supported by relational databases, such as now() and count(1). ◦ If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure MySQL Reader by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

The following sample code provides examples on how to configure a synchronization node to read data from a table that is not sharded and how to configure a synchronization node to read data from a sharded table.

- Configure a synchronization node to read data from a table that is not sharded

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"mysql",// The reader type.
            "parameter":{
                "column":[// The names of the columns from which you want to read data.
                    "id"
                ],
                "connection":[
                    {   "querySql":["select a,b from join1 c join join2 d on c.id = d.id;"], // The SQL state
ment that is used to read data from the source table.
                        "datasource":"",// The name of the data source.
                        "table":[// The name of the source table. The table name must be enclosed in brackets
[].
                            "xxx"
                        ]
                    }
                ],
                "where":"",// The WHERE clause.
                "splitPk":"",// The shard key.
                "encoding":"UTF-8"// The encoding format.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates th
at bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The m
bps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1,// The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

- Configure a synchronization node to read data from a sharded table

> **Note** When you configure a synchronization node to read data from a sharded MySQL table, you can select multiple partitions that have the same schema.

```
{
    "type": "job",
    "version": "1.0",
    "configuration": {
        "reader": {
            "plugin": "mysql",
            "parameter": {
                "connection": [
                    {
                        "table": [
                            "tbl1",
                            "tbl2",
                            "tbl3"
                        ],
                        "datasource": "datasourceName1"
                    },
                    {
                        "table": [
                            "tbl4",
                            "tbl5",
                            "tbl6"
                        ],
                        "datasource": "datasourceName2"
                    }
                ],
                "singleOrMulti": "multi",
                "splitPk": "db_id",
                "column": [
                    "id", "name", "age"
                ],
                "where": "1 < id and id < 100"
            }
        },
        "writer": {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
        "setting":{
            "errorLimit":{
               "record":"0"// The maximum number of dirty data records allowed.
        },
            "speed":{
               "throttle":false,// Specifies whether to enable bandwidth throttling. The value false indicate
s that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. T
he mbps parameter takes effect only when the throttle parameter is set to true.
               "concurrent":1,// The maximum number of parallel threads.
        }
    },
         "order":{
            "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
    }
}
```

5.2.8. Oracle Reader

This topic describes the data types and parameters that are supported by Oracle Reader and how to configure Oracle Reader by using the codeless user interface (UI) and code editor.

Oracle Reader can read data from Oracle.

Note

- ApsaraDB RDS and DRDS do not support Oracle.
- Oracle Reader uses the ojdbc7-12.1.0.2.jar driver to connect to Oracle databases. For more information about the supported versions of Oracle JDBC drivers, see Oracle JDBC FAQ.

Oracle Reader connects to a remote Oracle database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the statement on the database and returns data. Then, Oracle Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

- Oracle Reader generates the SQL statement based on the settings of the table, column, and where parameters and sends the statement to the Oracle database.
- If you set the querySql parameter, Oracle Reader sends the value of this parameter to the Oracle database.

Data types

Oracle Reader supports most Oracle data types. Make sure that the data types of your database are supported.

The following table describes the data types that are supported by Oracle Reader.

Category	Oracle data type
Integer	NUMBER, ROWID, INTEGER, INT, and SMALLINT
Floating point	NUMERIC, DECIMAL, FLOAT, DOUBLE PRECISION, and REAL
String	LONG, CHAR, NCHAR, VARCHAR, VARCHAR2, NVARCHAR2, CLOB, NCLOB, CHARACTER, CHARACTER VARYING, CHAR VARYING, NATIONAL CHARACTER, NATIONAL CHAR, NATIONAL CHARACTER VARYING, NATIONAL CHAR VARYING, and NCHAR VARYING
Date and time	TIMESTAMP and DATE
Boolean	BIT and BOOLEAN
Binary	BLOB, BFILE, RAW, and LONG RAW

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table from which you want to read data.	Yes	No default value

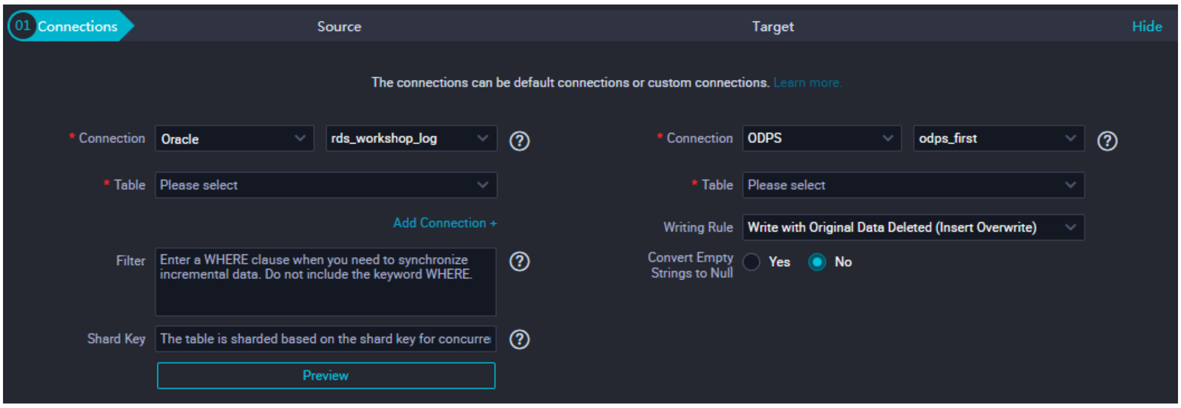
Parameter	Description	Required	Default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. The default value is ["*"], which indicates all the columns in the source table. - You can select specific columns to read. - The column order can be changed. This indicates that you can specify columns in an order different from the order specified by the schema of the source table. - Constants are supported. The column names must be arranged in JSON format. `["id", "1", "'mingya.wmy'", "null", "to_char(a + 1)", "2.3" , "true"]` ◦ id: a column name. ◦ 1: an integer constant. ◦ 'mingya.wmy': a string constant, which is enclosed in single quotation marks ('). ◦ null: a null pointer. ◦ to_char(a + 1): a function expression. ◦ 2.3: a floating-point constant. ◦ true: a Boolean value. - The column parameter cannot be left empty.	Yes	No default value
splitFactor	The shard factor, which determines the number of shards into which data to be synchronized is distributed. If you configure multiple parallel threads, the number of shards equals that the number of parallel threads multiplies by the value of the splitFactor parameter. For example, the number of parallel threads is 5, and the splitFactor parameter is set to 5. In this case, five parallel threads are used to perform sharding, and data is distributed into 25 shards. Note We recommend that you set this parameter in the range of 1 to 100. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	5
splitMode	The shard mode. Valid values: - **averageInterval**: average sampling. In this mode, the maximum and minimum values of all data are identified based on the **splitPk** parameter. Then, data is evenly distributed based on the number of shards. - **randomSampling**: random sampling. In this mode, data entries are randomly identified as sharding points. Note - If the splitPk parameter is set to a string field, set the **splitMode** parameter to **randomSampling**. - If the splitMode parameter is set to **averageInterval**, you can set the splitPk parameter only to a field of a numeric data type.	No	**randomSampling**

Parameter	Description	Required	Default value
splitPk	The field that is used for data sharding when Oracle Reader reads data. If you configure this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This improves data synchronization efficiency. • We recommend that you set the splitPk parameter to the name of the primary key column of the table. Data can be evenly distributed into different shards based on the primary key column, instead of being intensively distributed only into specific shards. • You can set the splitPk parameter to a field of any data type. • If you do not configure the splitPk parameter, Oracle Reader uses a single thread to read all data in the source table. **Note** If you use Oracle Reader to read data from a view, you cannot set the splitPk parameter to a field of the ROWID data type.	No	No default value
where	The WHERE clause. Oracle Reader generates an SQL statement based on the settings of the table, column, and where parameters and uses the statement to read data. For example, you can set this parameter to row_number() in a test. • You can use the WHERE clause to read incremental data. • If the where parameter is not provided or is left empty, Data Integration reads all data.	No	No default value
querySql (available only in the code editor)	The SQL statement that is used for refined data filtering. If you configure this parameter, Data Integration filters data based on the value of this parameter. For example, if you want to join multiple tables for data synchronization, you can set this parameter to `select a,b from table_a join table_b on table_a.id = table_b.id` . If you configure this parameter, Oracle Reader ignores the settings of the table, column, and where parameters.	No	No default value
fetchSize	The number of data records to read at a time. This parameter determines the number of interactions between Data Integration and the database and affects read efficiency. **Note** If you set this parameter to a value greater than 2048, an OOM error may occur during data synchronization.	No	1,024

Configure Oracle Reader by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



The connections can be default connections or custom connections. Learn more.

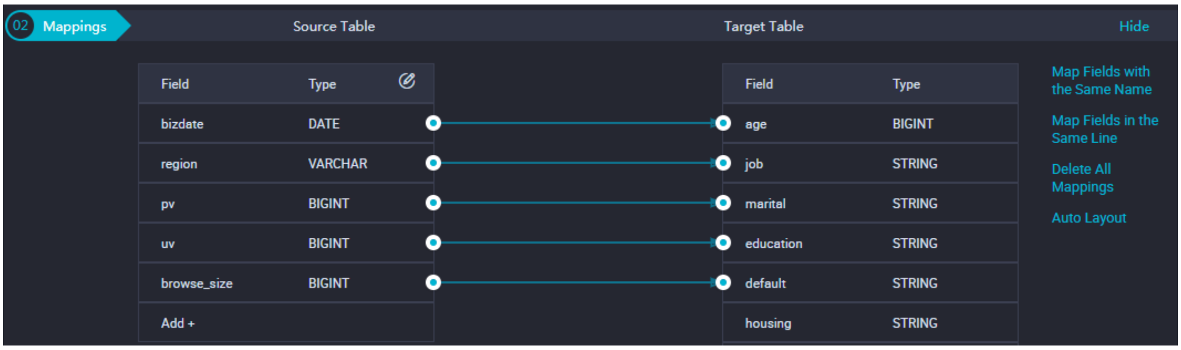
Source:
- * Connection: Oracle, rds_workshop_log
- * Table: Please select
- Add Connection +
- Filter: Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.
- Shard Key: The table is sharded based on the shard key for concurre

Target:
- * Connection: ODPS, odps_first
- * Table: Please select
- Writing Rule: Write with Original Data Deleted (Insert Overwrite)
- Convert Empty Strings to Null: ○ Yes ● No

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table from which you want to read data. This parameter is equivalent to the table parameter that is described in the preceding section.
Filter	The condition that is used to filter the data you want to read. Filtering based on the LIMIT keyword is not supported. The SQL syntax is determined by the selected data source.
Shard Key	The shard key. You can use a column in the source table as the shard key. We recommend that you use the primary key column or an indexed column. Only integer columns are supported by the codeless UI. If you want to use a column of other data types such as string, floating point, and date, use the code editor to configure Oracle Reader. If you configure this parameter, data sharding is performed based on the value of this parameter, and parallel threads can be used to read data. This improves data synchronization efficiency. **Note** The Shard Key parameter is displayed only after you select the data source for the synchronization node.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



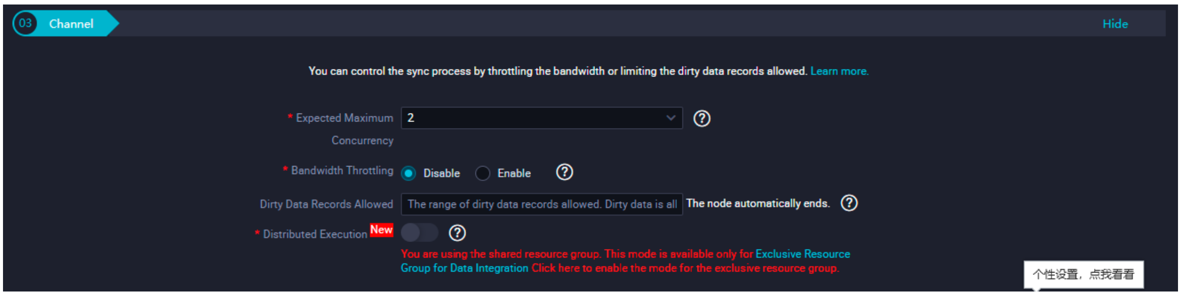
Field	Type
bizdate	DATE
region	VARCHAR
pv	BIGINT
uv	BIGINT
browse_size	BIGINT
Add +	

Field	Type
age	BIGINT
job	STRING
marital	STRING
education	STRING
default	STRING
housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout to sort the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.

Operation	Description
Add	◦ Click Add to add a field. Take note of the following rules when you add a field: You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ◦ You can use scheduling parameters, such as ${bizdate}. ◦ You can enter functions that are supported by relational databases, such as now() and count(1). ◦ If the field that you entered cannot be parsed, the value of Type for the field is **Unidentified**.

3. Configure channel control policies.



03 Channel

You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure Oracle Reader by using the code editor

In the following code, a synchronization node is configured to read data from an Oracle database:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"oracle",
            "parameter":{
                "fetchSize":1024,// The number of data records to read at a time.
                "datasource":"",// The name of the data source.
                "column":[// The names of the columns from which you want to read data.
                    "id",
                    "name"
                ],
                "where":"",// The WHERE clause.
                "splitPk":"",// The shard key.
                "table":""// The name of the table from which you want to read data.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1 // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
} "to":"Writer"
            }
        ]
    }
}
```

Additional information

- Data synchronization between primary and secondary databases

  A secondary Oracle database can be deployed for disaster recovery. The secondary database continuously synchronizes data from the primary database based on binary logs. Data latency between the primary and secondary databases cannot be prevented. This may result in data inconsistency.

- Data consistency control

Oracle is a relational database management system (RDBMS) that supports strong consistency for data queries. A database snapshot is created before a synchronization node starts. Oracle Reader reads data from the database snapshot. Therefore, if new data is written to the database during data synchronization, Oracle Reader cannot obtain the new data.

Data consistency cannot be ensured if you enable Oracle Reader to use parallel threads to read data in a synchronization node.

Oracle Reader shards the source table based on the value of the splitPk parameter and uses parallel threads to read data. These parallel threads belong to different transactions and read data at different points in time. Therefore, the parallel threads observe different snapshots.

Data inconsistencies cannot be prevented if parallel threads are used for a synchronization node. The following workarounds can be used:

  - Enable Oracle Reader to use a single thread to read data in a synchronization node. This indicates that you do not need to specify a shard key for Oracle Reader. This way, data consistency is ensured, but data is synchronized at low efficiency.
  - Make sure that no data is written to the source table during data synchronization. This ensures that the data in the source table remains unchanged during data synchronization. For example, you can lock the source table or disable data synchronization between primary and secondary databases. This way, data is efficiently synchronized, but your ongoing services may be interrupted.

- Character encoding

  Oracle Reader uses JDBC to read data. This enables Oracle Reader to automatically convert the encoding format of characters. Therefore, you do not need to specify the encoding format.

- Incremental data synchronization

  Oracle Reader connects to a database by using JDBC and uses a SELECT statement with a `WHERE` clause to read incremental data.

  - For batch data, incremental add, update, and delete operations (including logically delete operations) are distinguished by timestamps. Specify the WHERE clause based on a specific timestamp. The time indicated by the timestamp must be later than the time indicated by the latest timestamp in the previous synchronization.
  - For streaming data, specify the WHERE clause based on the ID of a specific record. The ID must be greater than the maximum ID involved in the previous synchronization.

  If the data that is added or modified cannot be distinguished, Oracle Reader can read only full data.

- Syntax validation

  Oracle Reader allows you to specify custom SELECT statements by using the querySql parameter but does not verify the syntax of these statements.

5.2.9. OSS Reader

This topic describes the data types and parameters that are supported by Object Storage Service (OSS) Reader and how to configure OSS Reader by using the codeless user interface (UI) and code editor.

OSS Reader reads data stored in OSS. OSS Reader uses Alibaba Cloud OSS SDK for Java to read data from OSS. Then, OSS Reader converts the data to a format that is readable to Data Integration and sends the converted data to a writer. OSS Reader supports the following OSS data types: BIGINT, DOUBLE, STRING, DATATIME, and BOOLEAN.

OSS stores only unstructured data. OSS Reader provides the following features:

- Reads data from TXT objects. The data in the objects must be logical two-dimensional tables.
- Reads data from CSV-like files with custom delimiters.
- Reads data of various types as strings and supports constants and column pruning.
- Supports recursive data read and object name-based filtering.
- Supports object compression. The following compression formats are supported: GZIP, BZIP2, and ZIP.

  > **Note** You cannot compress multiple objects into one package.

- Uses parallel threads to read data from multiple objects.

OSS Reader does not support the following features:

- Uses parallel threads to read data from a single object.
- Reads data from an object that exceeds 100 GB in size.

References:

- For more information about OSS, see What is OSS?
- For more information about OSS SDK for Java, see Alibaba Cloud OSS SDK for Java.
- For more information about how to process unstructured data such as data stored in OSS, see Access OSS data by using a built-in extractor.

Data types

Category	Data Integration data type	OSS data type
Integer	LONG	LONG
String	STRING	STRING
Floating point	DOUBLE	DOUBLE
Boolean	BOOLEAN	BOOLEAN
Date and time	DATE	DATE

Parameters

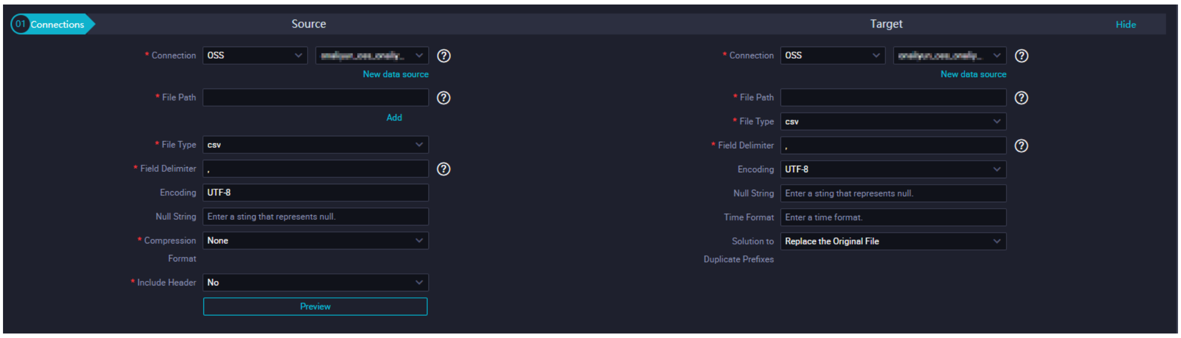
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
Object	The name of the OSS object from which you want to read data. You can specify multiple object names. For example, a bucket has a directory named yunshi, and this directory contains an object named ll.txt. In this case, you can set this parameter to yunshi/ll.txt. • If you specify a single OSS object name, OSS Reader uses only a single thread to read data. The feature of using parallel threads to read data from a single uncompressed object will be available in the future. • If you specify multiple OSS object names, OSS Reader uses parallel threads to read data. You can configure the number of parallel threads based on your business requirements. • If you specify a name that contains a wildcard, OSS Reader reads data from all objects that match the name. For example, if you set this parameter to `abc*[0-9]`, OSS Reader reads data from objects such as `abc0`, `abc1`, `abc2`, and `abc3`. If you set this parameter to `abc?.txt`, OSS Reader reads data from objects whose names start with `abc`, end with `.txt`, and contain an arbitrary character between abc and .txt. We recommend that you do not use wildcards because an out of memory (OOM) error may occur. For more information, see What is OSS? Note • Data Integration considers all objects in a synchronization node as a single table. Make sure that all objects in each synchronization node use the same schema. • Control the number of objects stored in a directory. If a directory contains excessive objects, an OOM error may occur. In this case, store the objects in different directories before you synchronize data.	Yes	No default value

<table>
<tr><th>Parameter</th><th>Description</th><th>Required</th><th>Default value</th></tr>
<tr><td>column</td><td>
<p>The names of the columns from which you want to read data. The type parameter specifies the source data type. The index parameter specifies the ID of the column in the source table, starting from 0. The value parameter specifies the column value if the column is a constant column.</p>
<p>By default, OSS Reader reads all data as strings. You can specify the column parameter in the following format:</p>
<pre>
json
"column": ["*"]
</pre>
<p>You can also specify the column parameter in the following format:</p>
<pre>
json
"column":
    {
       "type": "long",
       "index": 0   // The first INT-type column in the object
from which you want to read data.
    },
    {
       "type": "string",
       "value": "alibaba" // The value of the current column.
In this code, the value is the constant alibaba.
    }
</pre>
<p>Note For the column parameter, you must specify the type parameter and specify either the index or value parameter.</p>
</td><td>Yes</td><td>No default value</td></tr>
<tr><td>fieldDelimiter</td><td>
<p>The column delimiter that is used in the OSS object from which you want to read data.</p>
<p>Note You must specify a column delimiter for OSS Reader. The default column delimiter is commas (,). If you do not specify the column delimiter, the default column delimiter is used.</p>
<p>If the delimiter is non-printable, enter a value encoded in Unicode, such as \u001b and \u007c.</p>
</td><td>Yes</td><td>,</td></tr>
<tr><td>compress</td><td>The format in which objects are compressed. By default, this parameter is left empty, which means that objects are not compressed. OSS Reader supports the following compression formats: GZIP, BZIP2, and ZIP.</td><td>No</td><td><i>No default value</i></td></tr>
<tr><td>encoding</td><td>The encoding format of the object from which you want to read data.</td><td>No</td><td><i>utf-8</i></td></tr>
<tr><td>nullFormat</td><td>The string that represents a null pointer. No standard strings can represent a null pointer in TXT files. You can use this parameter to define a string that represents a null pointer. For example, if you specify <code>nullFormat="null"</code> , OSS Reader considers <code>null</code> as a null pointer. You can use the following formula to escape empty strings: <code>\N=\\N</code> .</td><td>No</td><td>No default value</td></tr>
<tr><td>skipHeader</td><td>Specifies whether to skip the headers in a CSV-like object if the object has headers. The skipHeader parameter is unavailable for compressed objects.</td><td>No</td><td><i>false</i></td></tr>
<tr><td>csvReaderConfig</td><td>The configurations required to read CSV objects. The parameter value must match the MAP type. You can use a CSV object reader to read data from CSV objects. The CSV object reader supports multiple configurations. If no configuration is performed, the default settings are used.</td><td>No</td><td>No default value</td></tr>
</table>

Configure OSS Reader by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



01 Connections

Source

- * Connection: OSS | [redacted]
- New data source
- * File Path:
- Add
- * File Type: csv
- * Field Delimiter: ,
- Encoding: UTF-8
- Null String: Enter a sting that represents null.
- * Compression Format: None
- * Include Header: No
- Preview

Target

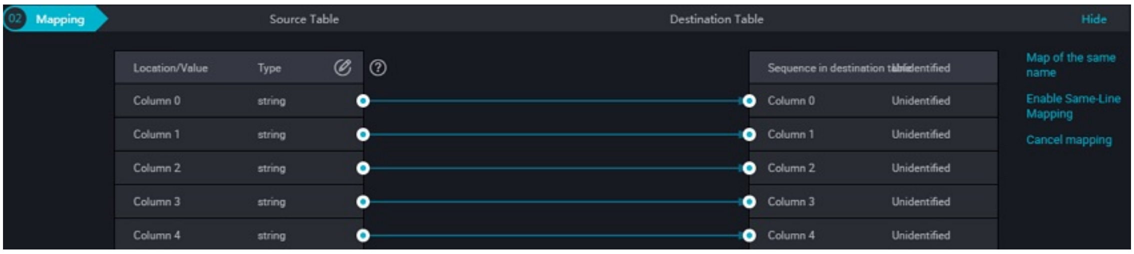
Hide

- * Connection: OSS | [redacted]
- New data source
- * File Path:
- * File Type: csv
- * Field Delimiter: ,
- Encoding: UTF-8
- Null String: Enter a sting that represents null.
- Time Format: Enter a time format.
- Solution to Duplicate Prefixes: Replace the Original File

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Object Name (Path Included)	The name of the object from which you want to read data. This parameter is equivalent to the Object parameter that is described in the preceding section. **Note** If an OSS object is named based on the date, such as aaa/20171024abc.txt, you can set this parameter to aaa/${bdp.system.bizdate}abc.txt.
Field Delimiter	The column delimiter. This parameter is equivalent to the fieldDelimiter parameter that is described in the preceding section. By default, a comma (,) is used as a column delimiter.
Encoding	This parameter is equivalent to the encoding parameter that is described in the preceding section. Default value: UTF-8.
Null String	The string that represents a null pointer. This parameter is equivalent to the nullFormat parameter that is described in the preceding section. If the source contains the string, the string is replaced with null.
Compression Format	The format in which objects are compressed. This parameter is equivalent to the compress parameter that is described in the preceding section. By default, objects are not compressed.
Include Header	Specifies whether to skip the headers in the object. This parameter is equivalent to the skipHeader parameter that is described in the preceding section. Default value: No.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



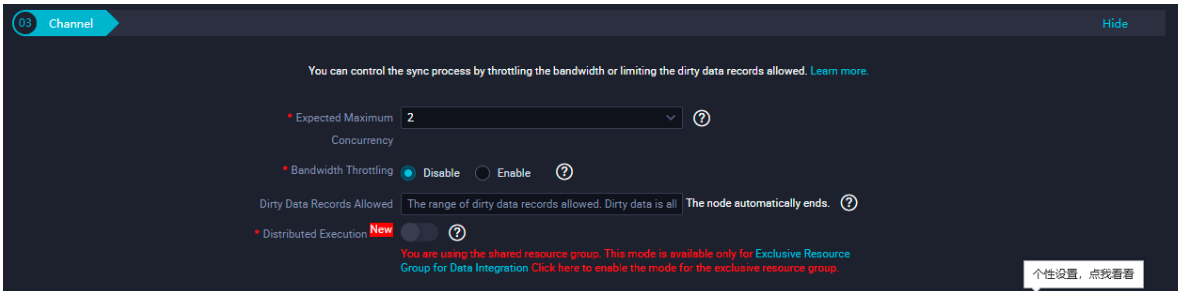
02 Mapping

Source Table | Destination Table | Hide

Location/Value	Type	Sequence in destination table	Unidentified
Column 0	string	Column 0	Unidentified
Column 1	string	Column 1	Unidentified
Column 2	string	Column 2	Unidentified
Column 3	string	Column 3	Unidentified
Column 4	string	Column 4	Unidentified

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.

3. Configure channel control policies.



03 Channel Hide

You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure OSS Reader by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to read data from OSS. For more information about parameters, see the preceding parameter description.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"oss",// The reader type.
            "parameter":{
                "nullFormat":"",// The string that represents a null pointer.
                "compress":"",// The format in which objects are compressed.
                "datasource":"",// The name of the data source.
                "column":[// The names of the columns from which you want to read data.
```

```
                    {
                        "index":0,// The ID of a column in the source object.
                        "type":"string"// The source data type.
                    },
                    {
                        "index":1,
                        "type":"long"
                    },
                    {
                        "index":2,
                        "type":"double"
                    },
                    {
                        "index":3,
                        "type":"boolean"
                    },
                    {
                        "format":"yyyy-MM-dd HH:mm:ss", // The time format.
                        "index":4,
                        "type":"date"
                    }
                ],
                "skipHeader":"",// Specifies whether to skip the headers in a CSV-like object if the object has
headers.
                "encoding":"",// The encoding format.
                "fieldDelimiter":",",// The column delimiter.
                "fileFormat": "",// The format of the object.
                "object":[]// The name of the object from which you want to read data.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":""// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1 // The maximum number of parallel threads.
            "mbps":"12",// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

Read data from ORC or Parquet objects in OSS

OSS Reader reads data from ORC or Parquet objects in the way in which HDFS Reader reads data. In addition to the original parameters, OSS Reader provides extended parameters such as Path and FileFormat.

- The following sample code provides an example on how to configure OSS Reader to read data from ORC objects in OSS:

```
{
      "stepType": "oss",
      "parameter": {
        "datasource": "",
        "fileFormat": "orc",
        "path": "/tests/case61/orc__691b6815_9260_4037_9899_****",
        "column": [
          {
            "index": 0,
            "type": "long"
          },
          {
            "index": "1",
            "type": "string"
          },
          {
            "index": "2",
            "type": "string"
          }
        ]
      }
    }
```

- The following sample code provides an example on how to configure OSS Reader to read data from Parquet objects in OSS:

```
{
"type": "job",
"version": "2.0",
"steps": [
{
"stepType": "oss",
"parameter": {
"nullFormat": "",
"compress": "",
"fileFormat": "parquet",
"path": "/*",
"parquetSchema": "message m { optional BINARY registration_dttm (UTF8); optional Int64 id; optional BINARY first_name (UTF8); optional BINARY last_name (UTF8); optional BINARY email (UTF8); optional BINARY gender (UTF8); optional BINARY ip_address (UTF8); optional BINARY cc (UTF8); optional BINARY country (UTF8); optional BINARY birthdate (UTF8); optional DOUBLE salary; optional BINARY title (UTF8); optional BINARY comments (UTF8); }",
"column": [
{
"index": "0",
"type": "string"
},
{
"index": "1",
"type": "long"
},
{
"index": "2",
"type": "string"
},
{
"index": "3",
"type": "string"
},
{
"index": "4",
"type": "string"
},
{
"index": "5",
"type": "string"
},
{
"index": "6",
"type": "string"
},
{
"index": "7",
"type": "string"
},
{
"index": "8",
"type": "string"
},
{
"index": "9",
"type": "string"
},
{
"index": "10",
"type": "double"
},
{
"index": "11",
"type": "string"
},
{
"index": "12",
"type": "string"
}
],
"skipHeader": "false",
"encoding": "UTF-8",
"fieldDelimiter": ",",
"fieldDelimiterOrigin": ",",
"datasource": "wpw_demotest_oss",
"envType": 0,
"object": [
"wpw_demo/userdata1.parquet"
]
},
"name": "Reader",
"category": "reader"
},
{
"stepType": "odps",
"parameter": {
"partition": "dt=${bizdate}",
"truncate": true,
"datasource": "0_odps_wpw_demotest",
"envType": 0,
"column": [
"id"
],
"emptyAsNull": false,
"table": "wpw_0827"
},
"name": "Writer",
"category": "writer"
}
],
"setting": {
"errorLimit": {
"record": ""
},
"locale": "zh_CN",
"speed": {
"throttle": false,
"concurrent": 2
}
},
"order": {
"hops": [
{
"from": "Reader",
"to": "Writer"
}
]
}
}
```

5.2.10. FTP Reader

This topic describes the data types and parameters that are supported by FTP Reader and how to configure FTP Reader by using the codeless user interface (UI) and code editor.

Background information

FTP Reader reads data from a remote FTP server. FTP Reader connects to a remote FTP server, reads data from the server, converts the data to a format that is readable to Data Integration, and then sends the data to a writer.

FTP Reader can read only FTP files that store logical two-dimensional tables, such as CSV files that store text data.

The files on the FTP server store only unstructured data. FTP Reader provides the following features:

- Reads data from TXT files. The data in the files must be logical two-dimensional tables.
- Reads data from CSV-like files with custom delimiters.
- Reads data of various types as strings and supports constants and column pruning.
- Supports recursive read and file name-based filtering.
- Supports file compression. The following compression formats are supported: GZIP, BZIP2, ZIP, LZO, and LZO_DEFLATE.
- Uses parallel threads to read data from multiple files.

FTP Reader cannot

- use parallel threads to read data from a single file.
- Uses concurrent threads to read a compressed file.

Data types

A remote FTP file does not distinguish between data types. The data types are defined by FTP Reader.

Data type in Data Integration	Data type in an FTP file
LONG	LONG
DOUBLE	DOUBLE
STRING	STRING
BOOLEAN	BOOLEAN
DATE	DATE

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	N/A
path	The path on the FTP server from which you want to read data. The path is a full path that contains the path of the directory and the file name with a suffix. You can specify multiple paths. - If you specify only one path, FTP Reader uses only one thread to read the related file. A feature of using parallel threads to read data from a single uncompressed file will be available in the future. - If you specify multiple paths, FTP Reader uses parallel threads to read the related files. The actual number of threads is determined by the number of channels. - If a path contains a wildcard, FTP Reader attempts to read all files that match the path. If a path ends with a forward slash (/), FTP Reader reads data from all files in the specified path. For example, if you specify the /bazhen/ path, FTP Reader reads data from all the files in the /bazhen directory. FTP Reader supports only asterisks (*) as wildcards. FTP Reader can flexibly generate node names based on custom parameters. Note - We recommend that you do not use asterisks (*) because an out of memory (OOM) error may occur on a Java Virtual Machine (JVM). - Data Integration considers all text files in a sync node as a single table. Make sure that all files in a sync node use the same schema. - Make sure that the data format is similar to CSV and readable to Data Integration. - If no readable files exist in the specified path, FTP Reader reports an error.	Yes	N/A

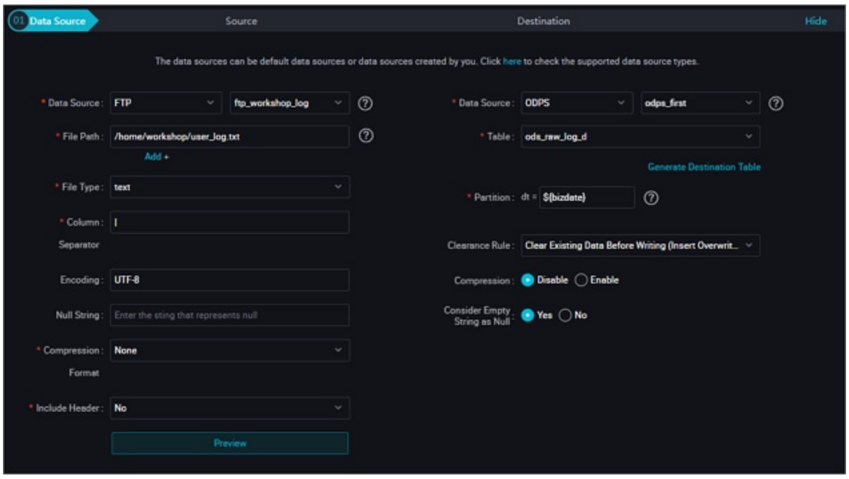
column	The columns from which you want to read data. The type parameter specifies the data type of a column. The index parameter specifies the ID of a column in the source table, starting from 0. The value parameter specifies the column value if the column is a constant column.
	By default, FTP Reader reads all data as strings. In this case, set this parameter to an asterisk (*), such as `"column":["*"]`. You can also set the column parameter in the following format:
	`{`
	`    "type": "long",`
	`    "index": 0    // The first INT-type column of the file from`
	`which you want to read data.`
	`  },`
	`  {`
	`    "type": "string",`
	`    "value": "alibaba"  // The value of the current column. In`
	`this code, the value is the constant "alibaba".`
	`  }`
	In the column parameter, you must specify the type parameter and specify either the index or value parameter.
		Yes	*

Parameter	Description	Required	Default value
fieldDelimiter	The column delimiter that is used in the file from which you want to read data. **Note** You must specify a column delimiter for FTP Reader. The default delimiter is commas (,). If you do not specify the column delimiter, the default column delimiter is used.	Yes	,
skipHeader	Specifies whether to skip the headers in a CSV-like file if the file contains headers. The skipHeader parameter is unavailable for compressed files. The default value of this parameter is false, which indicates that FTP Reader does not skip the headers in a CSV-like file.	No	false
encoding	The encoding format of the files that you want to write to the FTP server.	No	*utf-8*
nullFormat	The string that represents a null pointer. No standard strings can represent a null pointer in TXT files. You can use this parameter to define a string that represents a null pointer. For example, if you specify `nullFormat:"null"` , FTP Reader considers null as a null pointer.	No	N/A
markDoneFileName	The name of the file that is used to indicate that the sync node can start. Data Integration checks whether the file exists before data synchronization. If the file does not exist, Data Integration checks again later. Data Integration starts the sync node only after the file is detected.	No	N/A
maxRetryTime	The maximum number of retries for the detection of the file if no file is detected. By default, a maximum of 60 retries are allowed. Data Integration detects the file every 1 minute. The whole process lasts 60 minutes.	No	*60*
csvReaderConfig	The configurations required to read CSV files. The parameter value must match the MAP type. You can use a CSV file reader to read data from CSV files. The CSV file reader supports multiple configurations. If no configuration is performed, the default settings are used.	No	N/A
fileFormat	The format of the file. By default, FTP Reader reads data from CSV files. The data in CSV files must be logical two-dimensional tables. If you specify binary as the file format, data is converted to the binary format for replication and transmission. You can specify this parameter only when you want to replicate the complete directory structure between storage systems such as FTP and Object Storage Service (OSS).	No	N/A

Configure FTP Reader by using the codeless UI

1. Configure the source and destination.

Set parameters in the **Source** and **Target** sections for the sync node.



The data sources can be default data sources or data sources created by you. Click here to check the supported data source types.

Source:

- * Data Source: FTP, ftp_workshop_log
- * File Path: /home/workshop/user_log.txt
- * File Type: text
- * Column Separator: |
- Encoding: UTF-8
- Null String: Enter the sting that represents null
- * Compression Format: None
- * Include Header: No

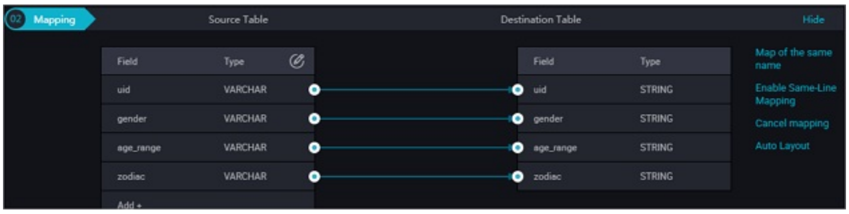
Destination:

- * Data Source: ODPS, odps_first
- * Table: ods_raw_log_d
- * Partition: dt = ${bizdate}
- Clearance Rule: Clear Existing Data Before Writing (Insert Overwrit...
- Compression: ● Disable ○ Enable
- Consider Empty String as Null: ● Yes ○ No

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
File Path	The path on the FTP server from which you want to read data. The path is a full path that contains the path of the directory and the file name with a suffix. This parameter is equivalent to the path parameter that is described in the preceding section.
File Type	The format of the file from which you want to read from the FTP server. The default format is CSV.
Field Delimiter	The column delimiter. This parameter is equivalent to the fieldDelimiter parameter that is described in the preceding section. By default, a comma (,) is used as a column delimiter.
Encoding	The encoding format. This parameter is equivalent to the encoding parameter that is described in the preceding section. Default value: *UTF-8*.
Null String	The string that represents a null pointer. This parameter is equivalent to the nullFormat parameter that is described in the preceding section.
Compression Format	The format in which files are compressed. By default, files are not compressed.
Skip Header	Specifies whether to skip the headers in the file. This parameter is equivalent to the skipHeader parameter that is described in the preceding section. Default value: *No*.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



Source Table:

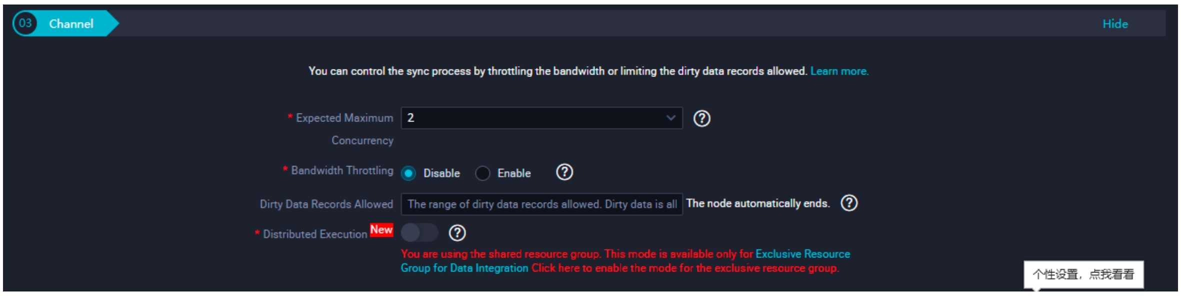
Field	Type
uid	VARCHAR
gender	VARCHAR
age_range	VARCHAR
zodiac	VARCHAR

Destination Table:

Field	Type
uid	STRING
gender	STRING
age_range	STRING
zodiac	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the sync node uses to read data from the source or write data to the destination. You can configure the parallelism for the sync node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure FTP Reader by using the code editor

In the following code, a sync node is configured to read data from an FTP server. For more information about how to configure a sync node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0", // The version number.
    "steps":[
        {
            "stepType":"ftp",// The reader type.
            "parameter":{
                "path":[],// The file path.
                "nullFormat":"",// The string that represents a null pointer.
                "compress":"",// The format in which files are compressed.
                "datasource":"", // The name of the data source.
                "column":[// The columns from which you want to read data.
                    {
                        "index":0,// The ID of the column from which you want to read data.
                        "type":""// The data type.
                    }
                ],
                "skipHeader":"",// Specifies whether to skip the headers in the file.
                "fieldDelimiter":",", // The column delimiter.
                "encoding":"UTF-8", // The encoding format.
                "fileFormat":"csv"// The format of the file.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0" // The maximum number of dirty data records allowed.
        },
        "speed":{
        "throttle": true, // Specifies whether to enable bandwidth throttling. A value of false indicates that bandwidth throttling is disabled, and a value of true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1 // The maximum number of parallel threads.
            "mbps":"12",// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.2.11. Tablestore Reader

This topic describes the data types and parameters that are supported by Tablestore Reader and how to configure Tablestore Reader by using the codeless user interface (UI) and code editor.

Tablestore Reader reads incremental data from Tablestore based on the specified range. Tablestore Reader reads incremental data in the following ways:

- Reads data from the entire table.

- Reads data based on the specified range.
- Reads data from the specified shard.

Tablestore is a NoSQL database service that is built on the Apsara distributed operating system and allows you to store and access large amounts of structured data in real time. Tablestore organizes data into instances and tables. It can seamlessly expand the data scale by using data sharding and load balancing technologies.

Tablestore Reader connects to the Tablestore server by using Tablestore SDK for Java and reads data from the server. Then, Tablestore Reader converts the data into a format that is readable to Data Integration based on the official data synchronization protocols, and sends the converted data to a writer.

Tablestore Reader splits a synchronization node into multiple concurrent tasks based on the table range to synchronize data in a Tablestore table. Each Tablestore Reader thread runs a task.

Tablestore Reader supports all Tablestore data types. The following table lists the data types supported by Tablestore Reader.

Category	Tablestore data type
Integer	INTEGER
Floating point	DOUBLE
String	STRING
Boolean	BOOLEAN
Binary	BINARY

Note Tablestore does not support DATE-type data. The application layer uses the LONG-type UNIX timestamp to indicate time.

Parameters

Parameter	Description	Required	Default value
endpoint	The endpoint of the Tablestore server. For more information, see Endpoints.	Yes	No default value
accessId	The AccessKey ID of the account that you use to connect to the Tablestore server.	Yes	No default value
accessKey	The AccessKey secret of the account that you use to connect to the Tablestore server.	Yes	No default value
instanceName	The name of the Tablestore instance. The instance is an entity for you to use and manage Tablestore. After you activate Tablestore, you must create an instance in the Tablestore console before you can create and manage tables. Instances are the basic units that you can use to manage Tablestore resources. Access control and resource metering for applications are implemented at the instance level.	Yes	No default value
table	The name of the table from which you want to read data. You can specify only one table. Multi-table synchronization is not required for Tablestore.	Yes	No default value

Parameter	Description	Required	Default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. Tablestore is a NoSQL database service. You must specify column names for Tablestore Reader to read data. • You can specify common columns. For example, you can specify {"name":"col1"} for Tablestore Reader to read data from column 1. • You can specify partial columns. Tablestore Reader reads only the specified columns. • You can specify constant columns. For example, you can specify {"type":"STRING", "value":"DataX"} for Tablestore Reader to read data from the column in which data is of the STRING type and the data value is DataX. The type parameter specifies the constant type. The supported types are STRING, INT, DOUBLE, BOOLEAN, BINARY, INF_MIN, and INF_MAX. If the constant type is BINARY, the constant value must be Base64-encoded. INF_MIN indicates the minimum value specified by Tablestore, and INF_MAX indicates the maximum value specified by Tablestore. If you set the type to INF_MIN or INF_MAX, do not set the value. If you set the value, errors may occur. • You cannot specify a function or custom expression. This is because Tablestore does not provide functions or expressions that are similar to those of SQL. Tablestore Reader cannot read data from columns that contain functions or expressions.	Yes	No default value

<table>
<tr><th>Parameter</th><th>Description</th><th>Required</th><th>Default value</th></tr>
<tr><td>begin and end</td><td>
<p>The Tablestore table range from which you want to read data. You must specify both or neither of the two parameters.</p>
<p>The begin and end parameters specify a range for primary key columns in the Tablestore table. Make sure that you specify a range for each primary key column in the table. If you do not need to limit a range, specify the parameters as <code>{"type":"INF_MIN"}</code> and <code>{"type":"INF_MAX"}</code>. The type parameter specifies the type of the data that you want to read.</p>
<p><b>Note</b></p>
<ul>
<li>Make sure that the number of primary keys is the same as the number of ranges indicated by begin and end. For example, the Tablestore table has n primary keys, and n is greater than or equal to 1. In this case, you must specify n ranges indicated by begin and end.</li>
<li>If the Tablestore table has multiple primary keys and the range specified for the first scanned primary key is <code>(INF_MIN,INF_MAX)</code>, Tablestore Reader does not scan other primary keys. Instead, it extracts all data from the table.</li>
</ul>
<p>For example, to read data from a Tablestore table with the primary keys of <code>[DeviceID, SellerID]</code>, specify the begin and end parameters in one of the following ways:</p>
<ul>
<li>Example 1:
<p>Extract <i>INT</i>-type data when the range specified for DeviceID is <code>(INF_MIN,INF_MAX)</code> and the range specified for SellerID is <code>(0,9999)</code>.</p>
<pre>
"range": {
      "begin": [
        {"type":"INF_MIN"}, // The minimum value of the DeviceID field.
        {"type":"INT", "value":"0"} // The minimum value of the SellerID field.
      ],
      "end": [
        {"type":"INF_MAX"}, // The maximum value of the DeviceID field.
        {"type":"INT", "value":"9999"} // The maximum value of the SellerID field.
      ]
    }
</pre></li>
<li>Example 2:
<p>Extract all data from the Tablestore table.</p>
<pre>
"range": {
      "begin": [
        {"type":"INF_MIN"}, // The minimum value of the DeviceID field.
        {"type":"INF_MIN"} // The minimum value of the SellerID field.
      ],
      "end": [
        {"type":"INF_MAX"}, // The maximum value of the DeviceID field.
          {"type":"INF_MAX"} // The maximum value of the SellerID field.
      ]
    }
</pre></li>
</ul>
</td><td>Yes</td><td>Left empty</td></tr>
</table>

<table>
<tr><th>Parameter</th><th>Description</th><th>Required</th><th>Default value</th></tr>
<tr><td>split</td><td>

The custom rule for data sharding. This parameter is an advanced configuration item. We recommend that you do not set this parameter.

If data is unevenly distributed in a Tablestore table and the automatic sharding feature of Tablestore Reader fails to work, you can customize a sharding rule.

The sharding rule that is specified by the split parameter must fall in the range that is specified by the begin and end parameters and must be the values of the partition key. This means that you specify only the values of the partition key instead of the values of all the primary key columns in the split parameter.

To read data from a Tablestore table with the primary keys of `[DeviceID, SellerID]` , specify the following parameters:

```
"range": {
      "begin": {
        {"type":"INF_MIN"},  // The minimum value of the DeviceID field.
        {"type":"INF_MIN"}  // The minimum value of the SellerID field.
      },
      "end": {
        {"type":"INF_MAX"}, // The maximum value of the DeviceID field.
        {"type":"INF_MAX"} // The maximum value of the SellerID field.
      },
       // The specified sharding rule. If you specify a sharding rule, the synchronization node is split into concurrent tasks based on the values of the begin, end, and split parameters. Data is sharded based only on the partition key, which is the first primary key column.
       // The data type of the partition key can be INF_MIN, INF_MAX, STRING, or INT.
            "split":[
                                {"type":"STRING", "value":"1"},
                                {"type":"STRING", "value":"2"},
                                {"type":"STRING", "value":"3"},
                                {"type":"STRING", "value":"4"},
                                {"type":"STRING", "value":"5"}
                    ]
    }
```

</td><td>No</td><td>No default value</td></tr>
</table>

Configure Tablestore Reader by using the codeless UI

This method is not supported.

Configure Tablestore Reader by using the code editor

In the following code, a synchronization node is configured to read data from a Tablestore table by using the code editor. Fore more information, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"ots",// The reader type.
            "parameter":{
                "datasource":"",// The name of the data source.
                "column":[// The names of the columns from which you want to read data.
                    {
```

```
                    {
                        "name":"column1"// The name of the column.
                    },
                    {
                        "name":"column2"
                    },
                    {
                        "name":"column3"
                    },
                    {
                        "name":"column4"
                    },
                    {
                        "name":"column5"
                    }
                ],
                "range":{
                    "split":[
                        {
                            "type":"INF_MIN"
                        },
                        {
                            "type":"STRING",
                            "value":"splitPoint1"
                        },
                        {
                            "type":"STRING",
                            "value":"splitPoint2"
                        },
                        {
                            "type":"STRING",
                            "value":"splitPoint3"
                        },
                        {
                            "type":"INF_MAX"
                        }
                    ],
                    "end":[
                        {
                            "type":"INF_MAX"
                        },
                        {
                            "type":"INF_MAX"
                        },
                        {
                            "type":"STRING",
                            "value":"end1"
                        },
                        {
                            "type":"INT",
                            "value":"100"
                        }
                    ],
                    "begin":[
                        {
                            "type":"INF_MIN"
                        },
                        {
                            "type":"INF_MIN"
                        },
                        {
                            "type":"STRING",
                            "value":"begin1"
                        },
                        {
                            "type":"INT",
                            "type":"INT",
                            "value":"0"
                        }
                    ]
                },
                "table":""// The name of the table from which you want to read data.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1 // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.2.12. AnalyticDB for MySQL 3.0 Reader

This topic describes the data types and parameters that are supported by AnalyticDB for MySQL 3.0 Reader and how to configure AnalyticDB for MySQL 3.0 Reader by using the codeless user interface (UI) and code editor.

AnalyticDB for MySQL 3.0 Reader reads data from tables stored in AnalyticDB for MySQL 3.0 databases. AnalyticDB for MySQL 3.0 Reader connects to a remote AnalyticDB for MySQL 3.0 database by using Java Database Connectivity (JDBC) and executes a SELECT statement to read data from the AnalyticDB for MySQL 3.0 database.

Data types

The following table lists the data types that are supported by AnalyticDB for MySQL 3.0 Reader.

Category	AnalyticDB for MySQL 3.0 data type
Integer	INT, INTEGER, TINYINT, SMALLINT, and BIGINT
Floating point	FLOAT, DOUBLE, and DECIMAL
String	VARCHAR
Date and time	DATE, DATETIME, TIMESTAMP, and TIME
Boolean	BOOLEAN

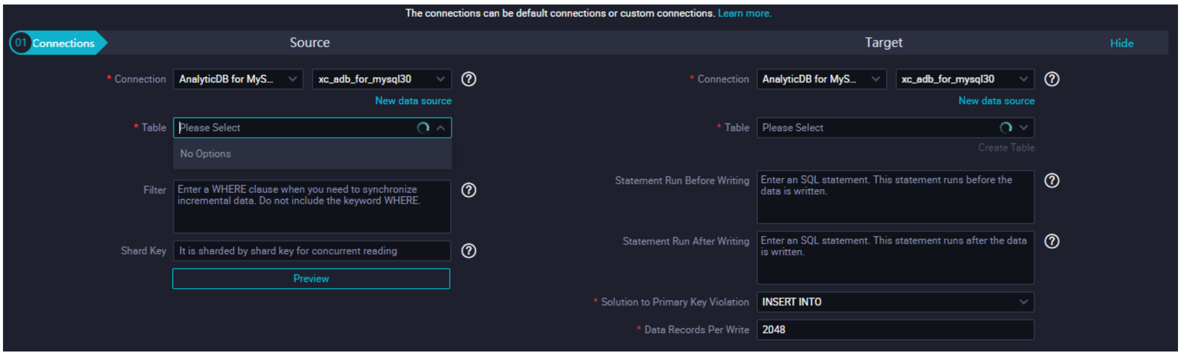
Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table from which you want to read data.	Yes	No default value
column	The names of the columns from which you want to read data. The columns are specified in a JSON array. The default value is [*], which indicates all columns. - You can select specific columns to read. - The column order can be changed. You can read data from the specified columns in an order different from that specified in the schema of the table. - Constants are supported. The column names must be arranged in compliance with the SQL syntax supported by MySQL, such as `` ["id", "`table`", "1", "'bazhen.csy'", "null", "to_char(a + 1)", "2.3" , "true"] ``. ○ id: a column name. ○ table: the name of a column that contains reserved keywords. ○ 1: an integer constant. ○ bazhen.csy: a string constant. ○ null: a null pointer. ○ to_char(a + 1): a function expression that is used to calculate the length of a string. ○ 2.3: a floating-point constant. ○ true: a Boolean value. - The column parameter must explicitly specify all the columns from which you want to read data. This parameter cannot be left empty.	Yes	No default value
splitPk	The field that is used for data sharding when AnalyticDB for MySQL 3.0 Reader reads data. If you specify this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This way, data can be synchronized more efficiently. - We recommend that you set the splitPk parameter to the name of the primary key column of the table. Data can be evenly distributed to different shards based on the primary key column, instead of being intensively distributed only to specific shards. - The splitPk parameter supports sharding for data only of integer data types. If you set this parameter to a field of an unsupported data type, such as a string, floating point, or date data type, AnalyticDB for MySQL 3.0 Reader ignores the splitPk parameter and uses a single thread to read data. - If the splitPk parameter is not provided or is left empty, AnalyticDB for MySQL 3.0 Reader uses a single thread to read data.	No	No default value
where	The WHERE clause. For example, you can set this parameter to `gmt_create > $bizdate` to read data on the current day. - You can use the WHERE clause to read incremental data. If the where parameter is not provided or is left empty, AnalyticDB for MySQL 3.0 Reader reads all data. - Do not set the where parameter to limit 10, which does not conform to the constraints of MySQL on the SQL WHERE clause.	No	No default value

Configure AnalyticDB for MySQL 3.0 Reader by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



The connections can be default connections or custom connections. Learn more.

Source:
- * Connection: AnalyticDB for MyS... / xc_adb_for_mysql30
- * Table: Please Select (No Options)
- Filter: Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.
- Shard Key: It is sharded by shard key for concurrent reading

Target:
- * Connection: AnalyticDB for MyS... / xc_adb_for_mysql30
- * Table: Please Select
- Statement Run Before Writing: Enter an SQL statement. This statement runs before the data is written.
- Statement Run After Writing: Enter an SQL statement. This statement runs after the data is written.
- * Solution to Primary Key Violation: INSERT INTO
- * Data Records Per Write: 2048

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table from which you want to read data. This parameter is equivalent to the table parameter that is described in the preceding section.
Filter	The condition used to filter the data that you want to read. Filtering based on the LIMIT keyword is not supported. The SQL syntax is determined based on the selected data source.
Shard Key	The shard key. You can use a column in the source table as the shard key. We recommend that you use the primary key column or an indexed column. Only integer columns are supported. If you specify this parameter, data sharding is performed based on the value of this parameter, and parallel threads can be used to read data. This improves data synchronization efficiency. **Note** The Shard Key parameter is displayed only after you select the data source for the synchronization node.

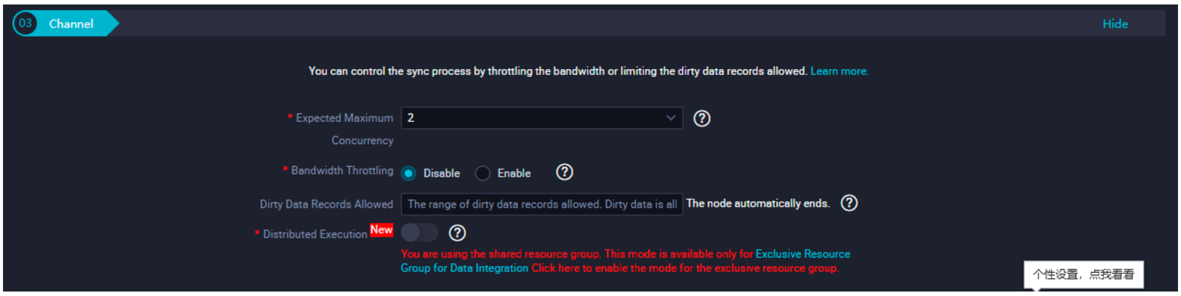
2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.

Operation	Description
Add	○ You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ○ You can use scheduling parameters such as ${bizdate}. ○ You can enter functions that are supported by relational databases, for example, now() and count(1). ○ If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure AnalyticDB for MySQL 3.0 Reader by using the code editor

In the following code, a synchronization node is configured to read data from AnalyticDB for MySQL 3.0. For more information about the parameters, see the preceding parameter description.

```
{
    "type": "job",
    "steps": [
        {
            "stepType": "analyticdb_for_mysql", // The reader type.
            "parameter": {
                "column": [ // The names of the columns from which you want to read data.
                    "id",
                    "value",
                    "table"
                ],
                "connection": [
                    {
                        "datasource": "xxx", // The name of the data source.
                        "table": [ // The name of the table from which you want to read data.
                            "xxx"
                        ]
                    }
                ],
                "where": "", // The condition used to filter data that you want to read.
                "splitPk": "", // The shard key.
                "encoding": "UTF-8" // The encoding format.
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "stream",
            "parameter": {},
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0" // The maximum number of dirty data records allowed.
        },
        "speed": {
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
                        "concurrent":1 // The maximum number of parallel threads.
                      "mbps":"12"// The maximum transmission rate.
        }
    }
}
```

5.2.13. ClickHouse Reader

This topic describes the parameters that are supported by ClickHouse Reader and how to configure ClickHouse Reader by using the codeless user interface (UI) and code editor.

? **Note** Only ApsaraDB for ClickHouse data sources are supported.

ClickHouse Reader reads data from tables stored in ClickHouse databases. ClickHouse Reader connects to a remote ApsaraDB for ClickHouse database by using Java Database Connectivity (JDBC) and executes SQL statements to read data from the ApsaraDB for ClickHouse database.

Limits

- ClickHouse Reader supports only exclusive resource groups for Data Integration, but not shared resource groups or custom resource groups for Data Integration. For more information, see Overview, Shared resource groups and resource plans for shared resource groups, and Overview.
- ClickHouse Reader connects to an ApsaraDB for ClickHouse database by using JDBC and can read data from a source table only by using JDBC Statement.
- ClickHouse Reader allows you to read data from the specified columns in an order different from that specified in the schema of the source table.
- You must make sure that the driver version is compatible with your ClickHouse database. ClickHouse Reader supports only the following version of the ClickHouse database driver:

```
<dependency>
    <groupId>ru.yandex.clickhouse</groupId>
    <artifactId>clickhouse-jdbc</artifactId>
    <version>0.2.4.ali2-SNAPSHOT</version>
</dependency>
```

Background information

ClickHouse Reader is designed for extract, transform, load (ETL) developers to read data from ApsaraDB for ClickHouse databases. ClickHouse Reader connects to a remote ApsaraDB for ClickHouse database by using JDBC, generates an SQL statement based on your configurations to read data from the database, matches the protocol of each writer of Data Integration, and writes data to other engines by using a write API that is provided by each engine.

Parameters

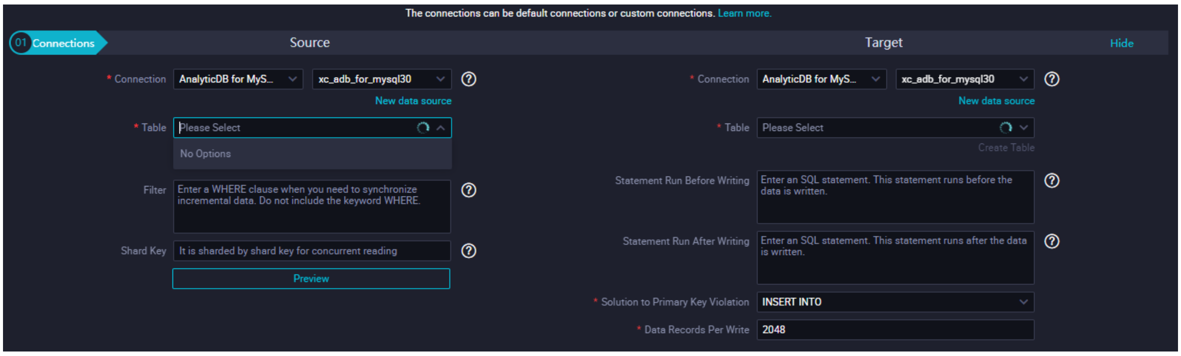
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table from which you want to read data. The table contains data in the JSON format. **Note** The table parameter must be included in the connection parameter.	Yes	No default value
fetchSize	The number of data records to read at a time. This parameter determines the number of interactions between Data Integration and the source database and affects read efficiency. **Note** If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization. You can increase the value of this parameter based on the workloads on ClickHouse.	No	1,024
column	The names of the columns from which you want to read data. Separate the names with commas (,). Example: "column": ["id", "name", "age"]. **Note** The column parameter cannot be left empty.	Yes	No default value

Parameter	Description	Required	Default value
jdbcUrl	The JDBC URL of the source database. The jdbcUrl parameter must be included in the connection parameter. • You can configure only one JDBC URL for a database. • The value format of the jdbcUrl parameter must be in accordance with the official specifications of ClickHouse. You can also specify additional JDBC connection properties in the value of this parameter. Example: jdbc:clickhouse://localhost:3306/test?user=root&password=&useUnicode=true&characterEncoding=gbk&autoReconnect=true&failOverReadOnly=false.	Yes	No default value
username	The username that you can use to connect to the database.	Yes	No default value
password	The password that you can use to connect to the database.	Yes	No default value
splitPk	The field that is used for data sharding when ClickHouse Reader reads data. If you specify this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This way, data can be synchronized more efficiently. **Note** If splitPk is specified, you must specify the fetchSize parameter.	No	No default value
where	The WHERE clause. For example, you can set this parameter to `gmt_create > $bizdate` to read the data that is generated on the current day. You can use the WHERE clause to read incremental data. If the where parameter is not provided or is left empty, ClickHouse Reader reads all data.	No	No default value

Configure ClickHouse Reader by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



The connections can be default connections or custom connections. Learn more.

Source

* Connection: AnalyticDB for MyS... xc_adb_for_mysql30

* Table: Please Select

No Options

Filter: Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.

Shard Key: It is sharded by shard key for concurrent reading

Target

* Connection: AnalyticDB for MyS... xc_adb_for_mysql30

* Table: Please Select

Statement Run Before Writing: Enter an SQL statement. This statement runs before the data is written.

Statement Run After Writing: Enter an SQL statement. This statement runs after the data is written.

* Solution to Primary Key Violation: INSERT INTO

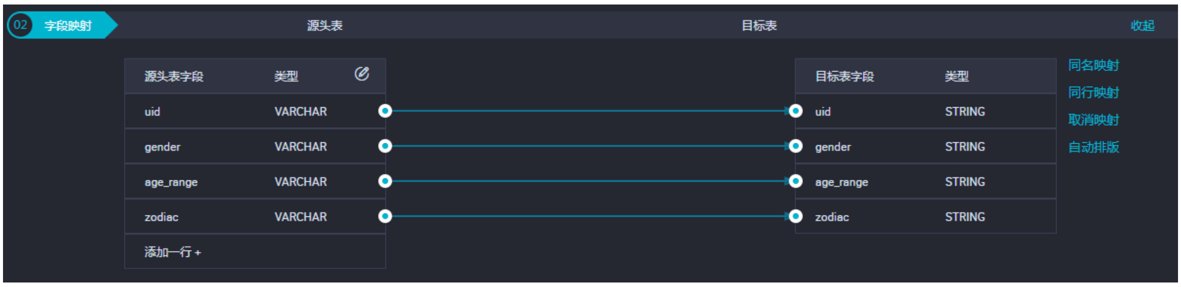
* Data Records Per Write: 2048

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is specified in the preceding section.
Table	The name of the table from which you want to read data. This parameter is equivalent to the table parameter that is specified in the preceding section.

Parameter	Description
Filter	The WHERE clause. This parameter is equivalent to the where parameter that is specified in the preceding section. The condition that is used to filter the data you want to read. Filtering based on the LIMIT keyword is not supported. The SQL syntax is determined based on the selected data source.
Shard Key	The shard key. This parameter is equivalent to the splitPk parameter that is specified in the preceding section. You can use a column in the source table as the shard key. We recommend that you use the primary key column or an indexed column. Only integer columns are supported. If you specify this parameter, data sharding is performed based on the value of this parameter, and parallel threads can be used to read data. This improves data synchronization efficiency. **Note** The Shard Key parameter is displayed only after you select the data source for the synchronization node.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.

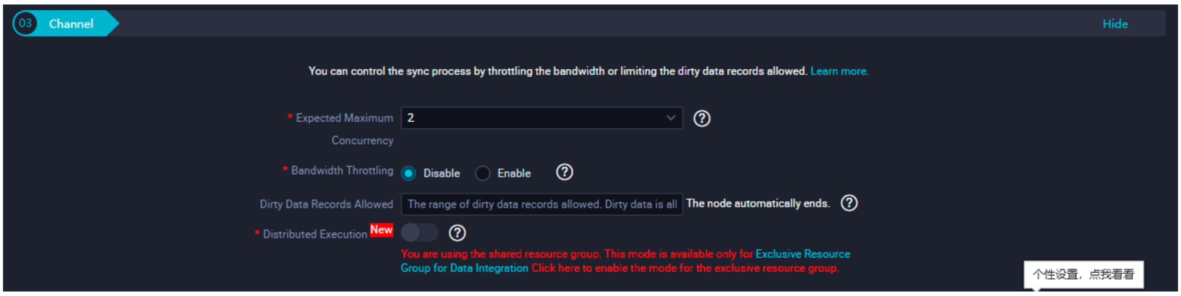


源头表字段	类型
uid	VARCHAR
gender	VARCHAR
age_range	VARCHAR
zodiac	VARCHAR
添加一行 +	

目标表字段	类型
uid	STRING
gender	STRING
age_range	STRING
zodiac	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source and destination tables. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	Click Add to add a field. Take note of the following rules when you add a field: ◦ You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ◦ You can use scheduling parameters, such as ${bizdate}. ◦ You can enter functions that are supported by relational databases, such as now() and count(1). ◦ If the field that you entered cannot be parsed, the value of Type for the field is **Unidentified**.

3. Configure channel control policies.



03 Channel

Hide

You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure ClickHouse Reader by using the code editor

In the following code, a synchronization node is configured to read data from ApsaraDB for ClickHouse. For more information about the parameters, see the preceding parameter description.

> **Note** Delete the comments from the code before you run the code.

```
{
    "type": "job",
    "version": "2.0",
    "steps": [
        {
            "stepType": "clickhouse", // The reader type.
            "parameter": {
                "fetchSize":1024,// The number of data records to read at a time.
                "datasource": "example",
                "column": [   // The names of the columns from which you want to read data.
                    "id",
                    "name"
                ],
                "where": "",    // The condition that is used to filter data you want to read.
                "splitPk": "",  // The shard key.
                "table": ""    // The name of the table from which you want to read data.
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "clickhouse",
            "parameter": {
                "postSql": [
                    "update @table set db_modify_time = now() where db_id = 1"
                ],
                "datasource": "example",    // The name of the data source.
                "batchByteSize": "67108864",
                "column": [
                    "id",
                    "name"
                ],
                "writeMode": "insert",
                "encoding": "UTF-8",
                "batchSize": 1024,
                "table": "ClickHouse_table",
                "preSql": [
                    "delete from @table where db_id = -1"
                ]
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "setting": {
        "executeMode": null,
        "errorLimit": {
            "record": "0"  // The maximum number of dirty data records allowed.
        },
        "speed": {
         "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that ba
ndwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps par
ameter takes effect only when the throttle parameter is set to true.
            "concurrent":1 // The maximum number of parallel threads.
            "mbps":"12",// The maximum transmission rate.
        }
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}
```

5.2.14. SQL Server Reader

This topic describes the data types and parameters that are supported by SQL Server Reader and how to configure SQL Server Reader by using the codeless user interface (UI) and code editor.

SQL Server Reader reads data from SQL Server.

SQL Server Reader connects to a remote SQL Server database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the statement on the database and returns data. Then, SQL Server Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer. If you use the code editor to configure SQL Server Reader, take note of the following items:

- SQL Server Reader generates the SQL statement based on the settings of the table, column, and where parameters and sends the generated statement to the SQL Server database.
- If you specify the querySql parameter, SQL Server Reader sends the value of this parameter to the SQL Server database.

SQL Server Reader supports most SQL Server data types. Make sure that the data types of your database are supported.

SQL Server versions

SQL Server Reader uses the driver com.microsoft.sqlserver sqljdbc4 4.0. For more information about the capabilities of the driver, see the official documentation. The following table lists the commonly used SQL Server versions and describes whether they are supported by the driver.

Version	Supported
SQL Server 2016	Yes
SQL Server 2014	Yes
SQL Server 2012	Yes
PDW 2008R2 AU34	Yes
SQL Server 2008 R2	Yes
SQL Server 2008	Yes
SQL Server 2019	No
SQL Server 2018	No

Data types

The following table lists the data types that are supported by SQL Server Reader.

Category	SQL Server data type
Integer	BIGINT, INT, SMALLINT, and TINYINT
Floating point	FLOAT, DECIMAL, REAL, and NUMERIC
String	CHAR, NCHAR, NTEXT, NVARCHAR, TEXT, VARCHAR, NVARCHAR (MAX), and VARCHAR (MAX)
Date and time	DATE, DATETIME, and TIME
Boolean	BIT
Binary	BINARY, VARBINARY, VARBINARY (MAX), and TIMESTAMP

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table from which you want to read data. Each synchronization node can be used to synchronize data to only one table.	Yes	No default value

Parameter	Description	Required	Default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. The default value is [*], which indicates all the columns in the source table. - You can select specific columns to read. - The column order can be changed. This indicates that you can specify columns in an order different from the order specified by the schema of the source table. - Constants are supported. The column names must be arranged in compliance with the SQL syntax supported by SQL Server, such as `["id", "table","1", "'mingya.wmy'", "'null'", "to_char(a+1)", "2.3" , "true"]` . ○ id: a column name. ○ table: the name of a column that contains reserved keywords. ○ 1: an integer constant. ○ 'mingya.wmy': a string constant, which is enclosed in single quotation marks ('). ○ 'null': a string. ○ to_char(a + 1): a function expression. ○ 2.3: a floating-point constant. ○ true: a Boolean value. - The column parameter must explicitly specify all the columns from which you want to read data. The parameter cannot be left empty.	Yes	No default value
splitPk	The field that is used for data sharding when SQL Server Reader reads data. If you specify this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This way, data can be synchronized more efficiently. - We recommend that you set the splitPk parameter to the name of the primary key column of the table. Data can be evenly distributed to different shards based on the primary key column, instead of being intensively distributed only to specific shards. - The splitPk parameter supports sharding only for data of integer data types. If you set this parameter to a field of an unsupported data type, such as a string, floating point, or date data type, SQL Server Reader returns an error.	No	No default value
where	The WHERE clause. SQL Server Reader generates an SQL statement based on the settings of the column, table, and where parameters and uses the generated statement to read data. For example, when you perform a test, you can set the where parameter to limit 10. To read the data that is generated on the current day, you can set the where parameter to `gmt_create > $bizdate` . - You can use the WHERE clause to read incremental data. - If you do not specify the where parameter, SQL Server Reader reads all data.	No	No default value
querySql	The SQL statement that is used for refined data filtering. Specify this parameter in the format of `"querysql" : "SQL statement",` . If you specify this parameter, data is filtered based only on the value of this parameter. For example, if you want to join multiple tables for data synchronization, set this parameter to `select a,b from table_a join table_b on table_a.id = table_b.id` . If you specify this parameter, SQL Server Reader ignores the settings of the column, table, and where parameters.	No	No default value

Parameter	Description	Required	Default value
fetchSize	The number of data records to read at a time. This parameter determines the number of interactions between Data Integration and the source database and affects read efficiency. **Note** If you set this parameter to a value greater than 2048, an out of memory (OOM) error may occur during data synchronization.	No	*1024*

Configure SQL Server Reader by using the codeless UI

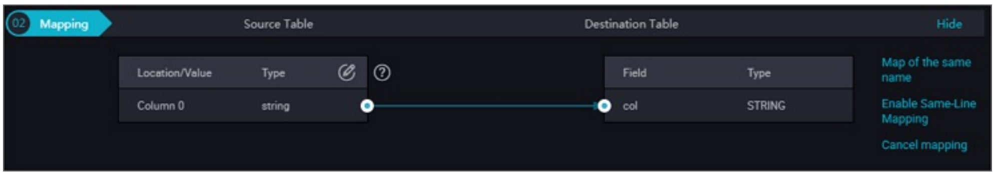
1. Configure data sources.

   Configure **Source** and Target for the synchronization node.

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table from which you want to read data. This parameter is equivalent to the table parameter that is described in the preceding section.
Filter	The condition that is used to filter the data you want to read. Filtering based on the LIMIT keyword is not supported. The SQL syntax is determined by the selected data source.
Shard Key	The shard key. You can use a column in the source table as the shard key. We recommend that you use the primary key column or an indexed column.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

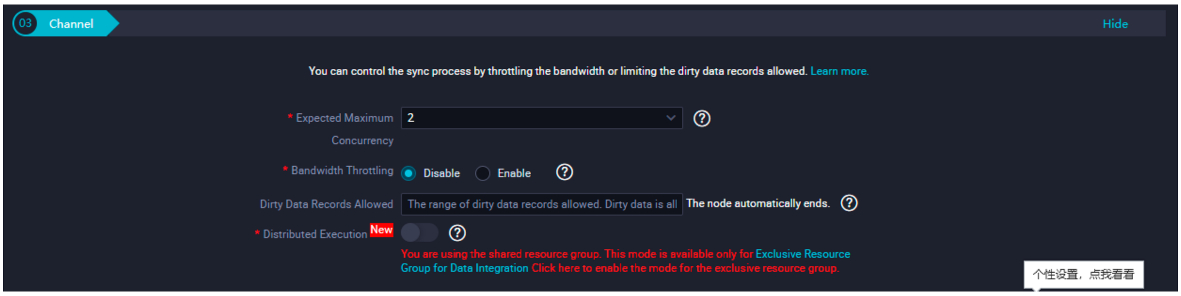
   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.

Operation	Description
Add	- Click Add to add a field. Take note of the following rules when you add a field: You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. - You can use scheduling parameters, such as ${bizdate}. - You can enter functions that are supported by relational databases, such as now() and count(1). - If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure SQL Server Reader by using the code editor

In the following code, a synchronization node is configured to read data from an SQL Server database:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"sqlserver",// The reader type.
            "parameter":{
                "datasource":"",// The name of the data source.
                "column":[// The names of the columns from which you want to read data.
                    "id",
                    "name"
                ],
                "where":"",// The WHERE clause.
                "splitPk":"",// The shard key based on which the table is sharded.
                "table":""// The name of the table from which you want to read data.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle": true,// Specifies whether to enable bandwidth throttling. The value false indicates tha
t bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps
parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1 // The maximum number of parallel threads.
            "mbps":"12",// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

You can use the querySql parameter to specify an SQL statement to read data. The following code provides an example. In the following code, sql_server_source is the SQL Server data source, dbo.test_table is the table from which you want to read data, and name is the column from which you want to read data.

```
{
    "stepType": "sqlserver",
    "parameter": {
        "querySql": "select name from dbo.test_table",
        "datasource": "sql_server_source",
        "column": [
            "name"
        ],
        "where": "",
        "splitPk": "id"
    },
    "name": "Reader",
    "category": "reader"
},
```

Additional information

- Data synchronization between primary and secondary databases

  A secondary SQL Server database can be deployed for disaster recovery. The secondary database continuously synchronizes data from the primary database based on binary logs. Data latency between the primary and secondary databases cannot be prevented. This may result in data inconsistency.

- Data consistency control

  SQL Server is a relational database management system (RDBMS) that supports strong consistency for data queries. A database snapshot is created before a synchronization node starts. SQL Server Reader reads data from the database snapshot. Therefore, if new data is written to the database during data synchronization, SQL Server Reader cannot obtain the new data.

  Data consistency cannot be ensured if you enable SQL Server Reader to use parallel threads to read data in a synchronization node.

  SQL Server Reader shards the source table based on the value of the splitPk parameter and uses parallel threads to read data. These parallel threads belong to different transactions and read data at different points in time. Therefore, the parallel threads observe different snapshots.

  Theoretically, data inconsistencies cannot be prevented if parallel threads are used for a synchronization node. The following workarounds can be used:

  - Enable SQL Server Reader to use a single thread to read data in a synchronization node. This indicates that you do not need to specify a shard key for SQL Server Reader. This way, data consistency is ensured, but data is synchronized at low efficiency.
  - Make sure that no data is written to the source table during data synchronization. This ensures that the data in the source table remains unchanged during data synchronization. For example, you can lock the source table or disable data synchronization between primary and secondary databases. This way, data can be efficiently synchronized, but your ongoing services may be interrupted.

- Character encoding

  SQL Server Reader uses JDBC to read data. This enables SQL Server Reader to automatically convert the encoding formats of characters. Therefore, you do not need to specify the encoding format.

- Incremental data synchronization

  SQL Server Reader uses JDBC to connect to a database and uses a SELECT statement with a `WHERE` clause to read incremental data.

  - For batch data, incremental add, update, and delete operations (including logically delete operations) are distinguished by timestamps. Specify the WHERE clause based on a specific timestamp. The time indicated by the timestamp must be later than the time indicated by the latest timestamp in the previous synchronization.
  - For streaming data, specify the WHERE clause based on the ID of a specific record. The ID must be greater than the maximum ID involved in the previous synchronization.

  If the data that is added or modified cannot be distinguished, SQL Server Reader can read only full data.

- Syntax validation

SQL Server Reader allows you to specify custom SELECT statements by using the querySql parameter but does not verify the syntax of these statements.

5.2.15. Lindorm Reader

This topic describes the data types and parameters that are supported by Lindorm Reader and how to configure Lindorm Reader by using the codeless user interface (UI) and code editor.

Context

Lindorm Reader reads data from tables stored in ApsaraDB for Lindorm databases. Lindorm Reader connects to a remote ApsaraDB for Lindorm database by using a Java client, and calls API operations to read data from the tables of the table and wideColumn types stored in the ApsaraDB for Lindorm database. Then, Lindorm Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

Note

- The configuration parameter is required for Lindorm Reader. You can go to the ApsaraDB for Lindorm console to obtain the configuration items that are necessary for Data Integration to connect to an ApsaraDB for Lindorm cluster. The configuration data must be in the JSON format.
- ApsaraDB for Lindorm is a multimode database. Lindorm Reader reads data from the tables of the table and wideColumn types stored in ApsaraDB for Lindorm databases. For more information about the tables of the table and wideColumn types, see Overview. You can also consult Lindorm engineers on duty by using DingTalk.

Limits

Lindorm Reader supports only exclusive resource groups for Data Integration, but not shared resource groups or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration, Use the default resource group, and Create a custom resource group for Data Integration.

Data types

Lindorm Reader supports most ApsaraDB for Lindorm data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by Lindorm Reader.

Category	ApsaraDB for Lindorm data type
Integer	INT, LONG, and SHORT
Floating point	DOUBLE, FLOAT, and DOUBLE
String	STRING
Date and time	DATE
Boolean	BOOLEAN
Binary	BINARYSTRING

Parameters

Parameter	Description	Required	Default value

Parameter	Description	Required	Default value
configuration	The configuration items that are necessary for Data Integration to connect to each ApsaraDB for Lindorm cluster. You can go to the ApsaraDB for Lindorm console to obtain the configuration items and ask the administrator of the ApsaraDB for Lindorm database to convert the configurations to data in the following JSON format: *{"key1":"value1","key2":"value2"}*. Example: *{"lindorm.zookeeper.quorum":"????","lindorm.zookeeper.property.clientPort":"????"}*. **Note** If you write the JSON code manually, you must escape double quotation marks (") of *value* to *\"*.	Yes	No default value
mode	The data read mode. Valid values: FixedColumn and DynamicColumn. Default value: FixedColumn.	Yes	FixedColumn
tablemode	The table type. Valid values: table and wideColumn. Default value: table. You can leave this parameter empty if a table of the table type is used.	No	This parameter is left empty by default.
table	The name of the table from which you want to read data. The table name is case-sensitive.	Yes	No default value
namespace	The namespace of the table from which you want to read data. The namespace of the table is case-sensitive.	Yes	No default value
encoding	The encoding method. Valid values: UTF-8 and GBK. This parameter is used to convert the lindorm byte[] data stored in binary mode to strings.	No	UTF-8

selects	Specifies whether to support parallel threads. If Lindorm Reader reads data from a table of the table type, parallel threads are not supported and a single synchronization node is run automatically. In this case, you must manually set the selects parameter. Example:
	`selects": [`
	`"where(compare(\"id\", LESS, 5))",`
	`"where(and(compare(\"id\", GREATER_OR_EQUAL, 5), compare(\"id\", LESS, 10)))",`
	`"where(compare(\"id\", GREATER_OR_EQUAL, 10))"`
	`],`
		No	No default value

<table>
<tr><th>Parameter</th><th>Description</th><th>Required</th><th>Default value</th></tr>
<tr><td>columns</td><td>The columns of the table from which you want to read data. Lindorm Reader allows you to read data from the specified columns in an order different from that specified in the schema of the source table.
<ul>
<li>If Lindorm Reader reads data from a table of the table type in an ApsaraDB for Lindorm database, you need only to specify the column names in a destination table. The column type information is automatically obtained based on the metadata of the source table and is filled in the destination table. Example:
<pre>
For a table of the table type:
[
    "id",
    "name",
    "age",
    "birthday",
    "gender"
]
</pre></li>
<li>Lindorm Reader reads data from a table of the wideColumn type stored in an ApsaraDB for Lindorm database. Example:
<pre>
For a table of the wideColumn type:
[
    "STRING|rowkey",
    "INT|f:a",
    "DOUBLE|f:b"
]
</pre></li>
</ul></td><td>Yes</td><td>No default value</td></tr>
</table>

Configure Lindorm Reader by using the codeless UI

This method is not supported.

Configure Lindorm Reader by using the code editor

- For more information about how to configure a job that reads data from a table of the table type stored in an ApsaraDB for Lindorm database to your server by using the code editor, see Create a synchronization node by using the code editor.

Note Delete the comments from the code before you run the code.

```
{
    "type": "job",
    "version": "2.0",
    "steps": [
        {
            "stepType": "lindorm",
            "parameter": {
                "mode": "FixedColumn",
            "caching": 128,
                "configuration": {    // The configuration items that are necessary for Data Integration to connect to each ApsaraDB for Lindorm cluster. The value is in the JSON format.
                    "lindorm.client.username": "",
                    "lindorm.client.seedserver": "seddserver.et2sqa.tbsite.net:30020",
                    "lindorm.client.namespace": "namespace",
                    "lindorm.client.password": ""
                },
                "columns": [
                    "id",
                    "name",
                    "age",
                    "birthday",
                    "gender"
                ],
                "envType": 1,
                "datasource": "_LINDORM",
                "namespace": "namespace",
                "table": "lindorm_table"
            },
            "name": "lindormreader",
            "category": "reader"
        },
        {
            "stepType": "mysql",
            "parameter": {
                "postSql": [],
                "datasource": "_IDB.TAOBAO",
                "session": [],
                "envType": 1,
                "columns": "columns": [
                    "id",
                    "name",
                    "age",
                    "birthday",
                    "gender"
                ],
         "selects": [
                    "where(compare(\"id\", LESS, 5))",
                    "where(and(compare(\"id\", GREATER_OR_EQUAL, 5), compare(\"id\", LESS, 10)))",
                    "where(compare(\"id\", GREATER_OR_EQUAL, 10))"
                ],
                "socketTimeout": 3600000,
                "guid": "",
                "writeMode": "insert",
                "batchSize": 1024,
                "encoding": "UTF-8",
                "table": "",
                "preSql": []
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "setting": {
        "jvmOption": "",
        "executeMode": null,
        "errorLimit": {
            "record": "0"
        },
        "speed": {
        // The transmission rate, in Byte/s. Data Integration runs to reach this rate as much as possible but
does not exceed it.
        "byte": 1048576
      }
      // The maximum number of dirty data records allowed.
      "errorLimit": {
        // The maximum number of dirty data records allowed. If the value of errorlimit is greater than the m
aximum value, an error is reported.
        "record": 0,
        // The maximum percentage of dirty data records. 1.0 indicates 100% and 0.02 indicates 2%.
        "percentage": 0.02
      }
    },
    "order": {
        "hops": [
            {
                "from": "Reader"
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}
```

- For more information about how to configure the job that reads data from a table of the wideColumn type in an ApsaraDB for Lindorm database to your server by using the code editor, see Create a synchronization node by using the code editor.

> Note Delete the comments from the code before you run the code.

```
{
    "type": "job",
    "version": "2.0",
    "steps": [
        {
            "stepType": "lindorm",
            "parameter": {
                "mode": "FixedColumn",
                "configuration": {  // The configuration items that are necessary for Data Integration to con
nect to each ApsaraDB for Lindorm cluster. The value is in the JSON format.
                    "lindorm.client.username": "",
                    "lindorm.client.seedserver": "seddserver.et2sqa.tbsite.net:30020",
                    "lindorm.client.namespace": "namespace",
                    "lindorm.client.password": ""
                },
                "columns": "columns": [
                   "STRING|rowkey",
                      "INT|f:a",
                      "DOUBLE|f:b"
                ],
                "envType": 1,
                "datasource": "_LINDORM",
                "namespace": "namespace",
                "table": "wideColumn"
            },
            "name": "lindormreader",
            "category": "reader"
        },
        {
            "stepType": "mysql",
            "parameter": {
                "postSql": [],
                "datasource": "_IDB.TAOBAO",
                "session": [],
                "envType": 1,
                "column": [
                    "id",
                    "value"
                ],
                "socketTimeout": 3600000,
                "guid": "",
                "writeMode": "insert",
                "batchSize": 1024,
                "encoding": "UTF-8",
                "table": "",
                "preSql": []
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "setting": {
        "jvmOption": "",
        "jvmOption": "",
        "executeMode": null,
        "errorLimit": {
            "record": "0"
        },
        "speed": {
        // The transmission rate, in Byte/s. Data Integration runs to reach this rate as much as possible but
does not exceed it.
        "byte": 1048576
      }
        // The maximum number of dirty data records allowed.
        "errorLimit": {
        // The maximum number of dirty data records allowed. If the value of errorlimit is greater than the m
aximum value, an error is reported.
        "record": 0,
        // The maximum percentage of dirty data records. 1.0 indicates 100% and 0.02 indicates 2%.
        "percentage": 0.02
      }
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}
```

5.2.16. LogHub (SLS) Reader

This topic describes the data types and parameters that are supported by LogHub (SLS) Reader and how to configure LogHub Reader by using the codeless user interface (UI) and code editor.

Background information

Log Service is an all-in-one real-time data logging service and allows you to collect, consume, deliver, query, and analyze log data. It comprehensively improves the capabilities to process and analyze large amounts of logs. LogHub (SLS) Reader consumes real-time log data in LogHub (SLS) by using Log Service SDK for Java, converts the data to a format that is readable to Data Integration, and then sends the converted data to a writer.

How it works

LogHub (SLS) Reader consumes real-time log data in LogHub (SLS) by using Log Service SDK for Java. The following code provides an example of the SDK:

```
<dependency>
    <groupId>com.aliyun.openservices</groupId>
    <artifactId>aliyun-log</artifactId>
    <version>0.6.7</version>
</dependency>
```

In Log Service, a Logstore is a basic unit that you can use to collect, store, and query log data. The read and write logs of a Logstore are stored in a shard. Each Logstore consists of several shards. Each shard is defined by a left-closed, right-open interval of MD5 values so that intervals do not overlap each other. The range indicated by all intervals covers all the allowed MD5 values. Each shard can independently provide services.

- Write: 5 MB/s, 2,000 times/s
- Read: 10 MB/s, 100 times/s

LogHub (SLS) Reader consumes log data in shards based on the following process in which the GetCursor and BatchGetLog API operations are called:

- Obtains a cursor based on a time range.
- Reads logs based on the cursor and step parameters and returns the next cursor.

- Keeps moving the cursor to consume logs.
- Splits the node based on shards and uses parallel threads to run the node.

Data types

The following table lists the mapping between the Data Integration data type and LogHub (SLS) data type.

Data Integration data type	LogHub (SLS) data type
STRING	STRING

Parameters

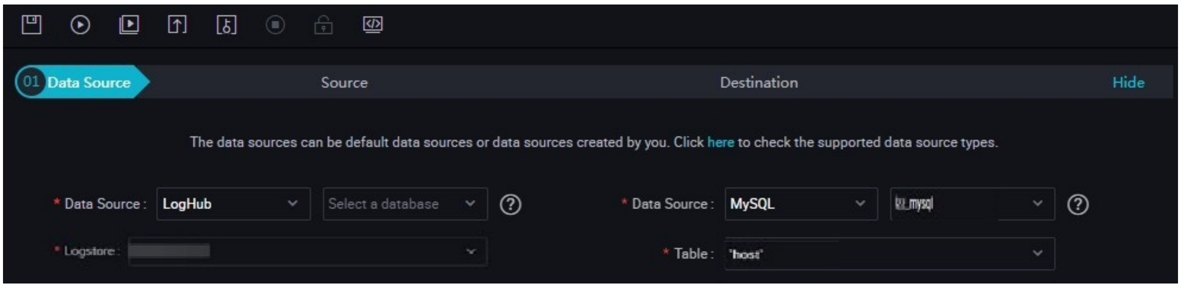
Parameter	Description	Required	Default value
endPoint	The endpoint of Log Service, which is a URL that you can use to access a project and log data. The endpoint varies based on the Alibaba Cloud region where the project resides and the project name. For more information about the endpoint of Log Service in each region, see Endpoints.	Yes	No default value
accessId	The AccessKey ID that is used to access Log Service.	Yes	No default value
accessKey	The AccessKey secret that is used to access Log Service.	Yes	No default value
project	The name of the project. A project is the basic unit for managing resources in Log Service. Projects are used to isolate resources and control access to the resources.	Yes	No default value
logstore	The name of the Logstore from which you want to read data. A Logstore is a basic unit that you can use to collect, store, and query log data in Log Service.	Yes	No default value
batchSize	The number of data entries that are queried from Log Service each time.	No	*128*
column	The names of the columns from which you want to read data. You can set this parameter to the metadata in Log Service. Supported metadata includes the log topic, unique identifier of the host, hostname, path, and log time. **Note** The column name is case-sensitive. For more information about column names in Log Service, see Introduction.	Yes	No default value
beginDateTime	The start time of data consumption. The value is the time at which log data reaches LogHub (SLS). This parameter defines the left boundary of a left-closed, right-open interval in the format of yyyyMMddHHmmss, such as 20180111013000. This parameter can work with the scheduling time parameter in DataWorks. For example, if you enter `beginDateTime=${yyyymmdd-1}` in the Parameters field on the **Properties** tab, you can set **Start Timestamp** to ${beginDateTime}000000 on the node configuration tab to consume logs that are generated from 00:00:00 of the data timestamp. **Note** The beginDateTime and endDateTime parameters must be used in pairs.	Yes	No default value

Parameter	Description	Required	Default value
endDateTime	The end time of data consumption. This parameter defines the right boundary of a left-closed, right-open interval in the format of yyyyMMddHHmmss, such as 20180111013010. This parameter can work with the scheduling time parameters in DataWorks. For example, if you enter endDateTime=${yyyymmdd} in the **Parameters** field on the Properties tab, you can set **End Timestamp** to ${endDateTime}000000 on the node configuration tab to consume logs that are generated until 00:00:00 of the next day of the data timestamp. **Note** The time that is specified by the endDateTime parameter of the previous interval cannot be earlier than the time that is specified by the beginDateTime parameter of the current interval. Otherwise, data in some regions may not be read.	Yes	No default value

Configure LogHub (SLS) Reader by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



The data sources can be default data sources or data sources created by you. Click here to check the supported data source types.

* Data Source : LogHub | Select a database

* Logstore : [redacted]

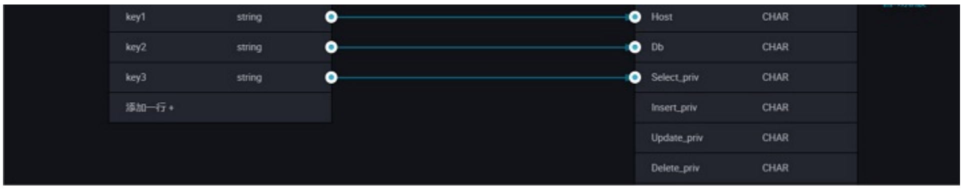
* Data Source : MySQL | [redacted]_mysql

* Table : 'host'

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Logstore	The name of the Logstore from which you want to read data.
Start Timestamp	The start time of data consumption. The value is the time at which log data reaches LogHub (SLS). This parameter defines the left boundary of a left-closed, right-open interval in the format of yyyyMMddHHmmss, such as 20180111013000. This parameter can work with the scheduling time parameter in DataWorks.
End Timestamp	The end time of data consumption. This parameter defines the right boundary of a left-closed, right-open interval in the format of yyyyMMddHHmmss, such as 20180111013010. This parameter can work with the scheduling time parameters in DataWorks.
Records per Batch	The number of data entries that are queried from Log Service each time.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. beginTimestampMillis

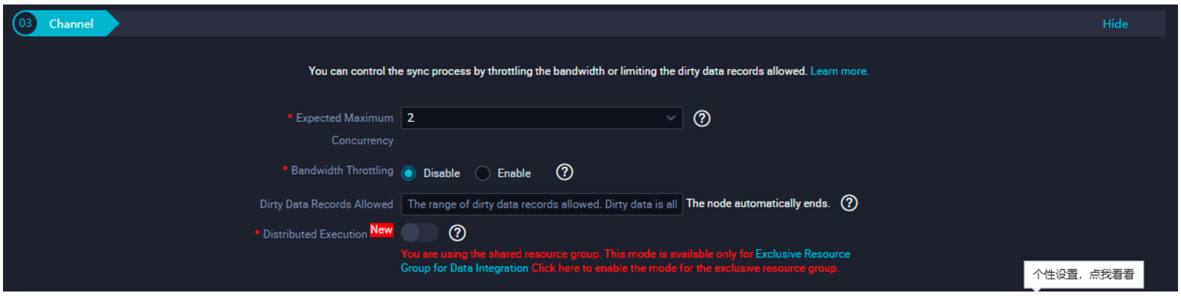
Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



key1 string — Host CHAR
key2 string — Db CHAR
key3 string — Select_priv CHAR
添加一行 + — Insert_priv CHAR
Update_priv CHAR
Delete_priv CHAR

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure LogHub (SLS) Reader by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to read data from LogHub (SLS). For more information about the parameters, see the preceding parameter description.

```
{
 "type":"job",
 "version":"2.0",// The version number.
 "steps":[
     {
         "stepType":"loghub",// The reader type.
         "parameter":{
             "datasource":"",// The name of the data source.
             "column":[// The names of the columns from which you want to read data.
                 "col0",
                 "col1",
                 "col2",
                 "col3",
                 "col4",
                 "C_Category",
                 "C_Source",
                 "C_Topic",
                 "C_MachineUUID", // The log topic.
                 "C_HostName", // The hostname.
                 "C_Path", // The path.
                 "C_LogTime" // The time when the event occurred.
             ],
             "beginDateTime":"",// The start time of data consumption.
             "batchSize":"",// The number of data entries that are queried from Log Service at a time.
             "endDateTime":"",// The end time of data consumption.
             "fieldDelimiter":",",// The column delimiter.
             "logstore":""// The name of the Logstore from which you want to read data.
         },
         "name":"Reader",
         "category":"reader"
     },
     {
         "stepType":"stream",
         "parameter":{},
         "name":"Writer",
         "category":"writer"
     }
 ],
 "setting":{
     "errorLimit":{
         "record":"0"// The maximum number of dirty data records allowed.
     },
     "speed":{
         "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that ba
ndwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps par
ameter takes effect only when the throttle parameter is set to true.
            "concurrent":1 // The maximum number of parallel threads.
            "mbps":"12",// The maximum transmission rate.
     }
 },
 "order":{
     "hops":[
         {
             "from":"Reader",
             "to":"Writer"
         }
     ]
 }
}
```

Note If the metadata in JSON format is prefixed by tag, delete the tag prefix. For example, change `__tag__:__client_ip__` to `__client_ip__` .

5.2.17. OTSReader-Internal

This topic describes the data types and parameters that are supported by OTSReader-Internal and how to configure OTSReader-Internal by using the codeless user interface (UI) and code editor.

Tablestore is a NoSQL database service that is built on the Apsara distributed operating system and allows you to store and access large amounts of structured data in real time. Tablestore organizes data into instances and tables. It can seamlessly expand the data scale by using data sharding and load balancing technologies.

OTSReader-Internal is used to export data from the Tablestore Internal model, and OTS Reader is used to export data from the Tablestore Public model.

The Tablestore Internal model supports the multi-version mode and normal mode. OTSReader-Internal can export data in the two modes:

- Multi-version mode: Tablestore stores multiple versions of column values, and this mode allows you to export data of multiple versions.

  OTSReader-Internal converts a cell to a 4-tuple of a one-dimensional table: PrimaryKey (columns 1 to 4), ColumnName, Timestamp, and Value. This process is similar to that for the multi-version mode of HBase Reader. Each {PrimaryKey, ColumnName, Timestamp, Value} tuple is sent to a writer as four columns in Data Integration records.

- Normal mode: This mode allows you to export the latest version of each column in each row, which is the same as the normal mode of HBase Reader. For more information, see the normal mode of HBase Reader in HBase Reader.

OTSReader-Internal connects to the Tablestore server and reads data by using official Tablestore SDK for Java. OTSReader-Internal provides some features, such as performing retry attempts when a timeout or exception occurs, to optimize the read process.

OTSReader-Internal supports all Tablestore data types. The following table lists the mappings between Data Integration data types and Tablestore data types.

Data Integration data type	Tablestore data type
LONG	INTEGER
DOUBLE	DOUBLE
STRING	STRING
BOOLEAN	BOOLEAN
BYTES	BINARY

Parameters

Parameter	Description	Required	Default value
mode	The mode in which OTSReader-Internal reads data. Valid values: *normal* and *multiVersion*.	Yes	No default value
endpoint	The endpoint of the Tablestore server.	Yes	No default value
accessId	The AccessKey ID that you use to access Tablestore.	Yes	No default value
accessKey	The AccessKey secret that you use to access Tablestore.	Yes	No default value

Parameter	Description	Required	Default value
instanceName	The name of the Tablestore instance. The instance is an entity for you to use and manage Tablestore. After you activate Tablestore, you must create an instance in the Tablestore console before you can create and manage tables. Instances are the basic units that you can use to manage Tablestore resources. Access control and resource measurement for applications are implemented at the instance level.	Yes	No default value
table	The name of the table from which you want to read data. You can specify only one table. Data synchronization to multiple tables is not supported for Tablestore.	Yes	No default value
range	The range of the data that you want to read, in the format of [begin,end). • If the value of the begin parameter is less than that of the end parameter, data is read in forward order. • If the value of the begin parameter is greater than that of the end parameter, data is read in reverse order. • The value of the begin parameter cannot be the same as that of the end parameter. • The following data types are supported for the type parameter: STRING, INT, and BINARY. Binary data is passed in as Base64 strings in the binary format. INF_MIN represents an infinitely small value, and INF_MAX represents an infinitely large value.	No	Whole table
range:{"begin"}	The start of the range of the data that you want to read. You can enter an empty array, a primary key prefix, or a complete primary key. If data is read in forward order, the default primary key suffix is INF_MIN. If data is read in reverse order, the default primary key suffix is INF_MAX. This parameter specifies the value range of the Tablestore primary key and is used for data filtering. If you do not specify this parameter, the minimum value is used by default. The JSON format does not support binary data. If the value type in the primary key column is BINARY, you must first use the Java method Base64.encodeBase64String to convert binary data to a string, and then enter the string as the value of the parameter. Example: • `byte[] bytes = "hello".getBytes();` : constructs binary data, which is the byte value of the string hello. • `String inputValue = Base64.encodeBase64String(bytes)` : calls the Base64.encodeBase64String method to convert the binary data to a string. After you run the preceding code, the string `"aGVsbG8="` is returned for the inputValue parameter. Finally, set this parameter to `{"type":"binary","value" : "aGVsbG8="}` .	No	Beginning of the table

Parameter	Description	Required	Default value
range:{"end"}	The end of the range of the data that you want to read. Enter an empty array, a primary key prefix, or a complete primary key. If data is read in forward order, the default primary key suffix is INF_MAX. If data is read in reverse order, the default primary key suffix is INF_MIN. The JSON format does not support binary data. If the value type in the primary key column is BINARY, you must use the Java method Base64.encodeBase64String to convert binary data to a string, and then enter the string as the value of the parameter. Example: • `byte[] bytes = "hello".getBytes();`: constructs binary data, which is the byte value of the string hello. • `String inputValue = Base64.encodeBase64String(bytes)`: calls the Base64.encodeBase64String method to convert the binary data to a string. After you run the preceding code, the string `"aGVsbG8="` is returned for the inputValue parameter. Finally, set this parameter to `{"type":"binary", "value":"aGVsbG8="}`.	No	End of the table
range:{"split"}	If you want to read large amounts of data, you can specify this parameter to split one node into multiple parallel tasks. Note • The value for the split parameter must be the shard key, which is the first column of the primary key, and the value type must be the same as that of the partition key. • The specified value must fall within the value range of the begin and end parameters. • The values for the split parameter must be sorted in descending or ascending order based on the data read order that is determined by the values of the begin and end parameters.	No	No default value
column	The names of the columns from which you want to read data. You can read data from regular and constant columns. Mode: The multi-version mode is supported. Format of regular columns: `{"name":"{your column name}"}`	Yes	No default value
timeRange (available only for the multi-version mode)	The time range of the data that is requested to read, in the format of [begin,end). Note The value of the begin parameter must be less than that of the end parameter.	No	All versions
timeRange:{"begin"} (available only for the multi-version mode)	The start time of the time range for reading data. Valid values: 0 to LONG_MAX.	No	0
timeRange:{"end"} (available only for the multi-version mode)	The end time of the time range for reading data. Valid values: 0 to LONG_MAX.	No	*LONG_MAX (9223372036854775806L)*

Parameter	Description	Required	Default value
maxVersion (available only for the multi-version mode)	The specified version of the requested data. Valid values: 1 to INT32_MAX.	No	All versions

Configure OTSReader-Internal Reader by using the codeless UI

This method is not supported.

Configure OTSReader-Internal Reader by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

- Multi-version mode

```
{
  "type": "job",
  "version": "2.0",
  "steps": [
       {
       "stepType": "ots-internal",
            "parameter": {
                "mode": "multiVersion",
                "endpoint": "",
                "accessId": "",
                "accessKey": "",
                "instanceName": "",
                "table": "",
                "range": {
                    "begin": [
                        {
                            "type": "string",
                            "value": "a"
                        },
                        {
                            "type": "INF_MIN"
                        }
                    ],
                    "end": [
                        {
                            "type": "string",
                            "value": "g"
                        },
                        {
                            "type": "INF_MAX"
                        }
                    ],
                    "split": [
                        {
                            "type": "string",
                            "value": "b"
                        },
                        {
                            "type": "string",
                            "value": "c"
                        }
                    ]
                },
                "column": [
                    {
                        "name": "attr1"
                    }
                ],
                "timeRange": {
                    "begin": 1400000000,
                    "end": 1600000000
                },
                "maxVersion": 10
            }
        }
    },
    "writer": {}
]
}
```

- Normal mode

```
{
  "type": "job",
  "version": "2.0",
  "steps":
 [
       {
           "stepType": "otsr-internal",
            "parameter": {
                "mode": "normal",
                "endpoint": "",
                "accessId": "",
                "accessKey": "",
                "instanceName": "",
                "table": "",
                "range": {
                    "begin": [
                        {
                            "type": "string",
                            "value": "a"
                        },
                        {
                            "type": "INF_MIN"
                        }
                    ],
                    "end": [
                        {
                            "type": "string",
                            "value": "g"
                        },
                        {
                            "type": "INF_MAX"
                        }
                    ],
                    "split": [
                        {
                            "type": "string",
                            "value": "b"
                        },
                        {
                            "type": "string",
                            "value": "c"
                        }
                    ]
                },
                "column": [
                    {
                        "name": "pk1"
                    },
                    {
                        "name": "pk2"
                    },
                    {
                        "name": "attr1"
                    },
                    {
                        "type": "string",
                        "value": ""
                    },
                    {
                        "type": "int",
                        "value": ""
                    },
                    {
                        "type": "double",
                        "value": ""
                    },
                    {
                        "type": "binary",
                        "value": "aGVsbG8="
                    }
                ]
            }
        }
    },
    "writer": {}
    ]
}
```

5.2.18. Stream Reader

This topic describes the data types and parameters that are supported by Stream Reader and how to configure Stream Reader by using the codeless user interface (UI) and code editor.

Stream Reader automatically generates data from the memory. It is mainly used to test the basic features and performance of data synchronization.

The following table lists the data types that are supported by Stream Reader.

Data type	Category
STRING	String
LONG	Long integer
DATE	Date and time
BOOLEAN	Boolean
BYTES	Byte

Parameters

<table>
<tr><th>Parameter</th><th>Description</th><th>Required</th><th>Default value</th></tr>
<tr><td>column</td><td>The data and types of columns in the source table. You can specify multiple columns. You can set this parameter to generate random strings of a specific length. The following code provides an example:
<pre>
"column" : [
      {
          "random": "8,15"
      },
      {
          "random": "10,10"
      }
]
</pre>
Configuration items:
<ul>
<li>"random": "8, 15": generates a random string that is 8 bytes to 15 bytes in length.</li>
<li>"random": "10, 10": generates a 10-byte random string.</li>
</ul></td><td>Yes</td><td>No default value</td></tr>
<tr><td>sliceRecordCount</td><td>The number of columns that are repeatedly generated.</td><td>Yes</td><td>No default value</td></tr>
</table>

Configure Stream Reader by using the codeless UI

This method is not supported.

Configure Stream Reader by using the code editor

In the following code, a synchronization node is configured to read data from the memory:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",// The reader type.
            "parameter":{
                "column":[// The data and types of columns in the source table.
                    {
                        "type":"string",// The data type.
                        "value":"field"// The value.
                    },
                    {
                        "type":"long",
                        "value":100
                    },
                    {
                        "dateFormat":"yyyy-MM-dd HH:mm:ss",// The time format.
                        "type":"date",
                        "value":"2014-12-12 12:12:12"
                    },
                    {
                        "type":"bool",
                        "value":true
                    },
                    {
                        "type":"bytes",
                        "value":"byte string"
                    }
                ],
                "sliceRecordCount":"100000"// The number of columns that are repeatedly generated.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1,// The maximum number of parallel threads.
           "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.2.19. HybridDB for MySQL Reader

This topic describes the data types and parameters that are supported by HybridDB for MySQL Reader and how to configure HybridDB for MySQL Reader by using the codeless user interface (UI) and code editor.

HybridDB for MySQL Reader can read tables and views. If you use HybridDB for MySQL Reader to read data from tables, you can specify all or some of the columns in the tables in sequence, change the column order, specify constant fields, and configure HybridDB for MySQL functions, such as now().

HybridDB for MySQL Reader reads data from HybridDB for MySQL.

HybridDB for MySQL Reader connects to a remote HybridDB for MySQL database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the SQL statement on the database and returns data. Then, HybridDB for MySQL Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

Data types

The following table lists the data types that are supported by HybridDB for MySQL Reader.

Category	HybridDB for MySQL data type
Integer	INT, TINYINT, SMALLINT, MEDIUMINT, and BIGINT
Floating point	FLOAT, DOUBLE, and DECIMAL
String	VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT
Date and time	DATE, DATETIME, TIMESTAMP, TIME, and YEAR
Boolean	BIT and BOOLEAN
Binary	TINYBLOB, MEDIUMBLOB, BLOB, LONGBLOB, and VARBINARY

Note

- Data types that are not listed in the preceding table are not supported.
- HybridDB for MySQL Reader processes TINYINT(1) as an integer data type.

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table from which you want to read data. Each synchronization node can be used to synchronize data to only one table.	Yes	No default value

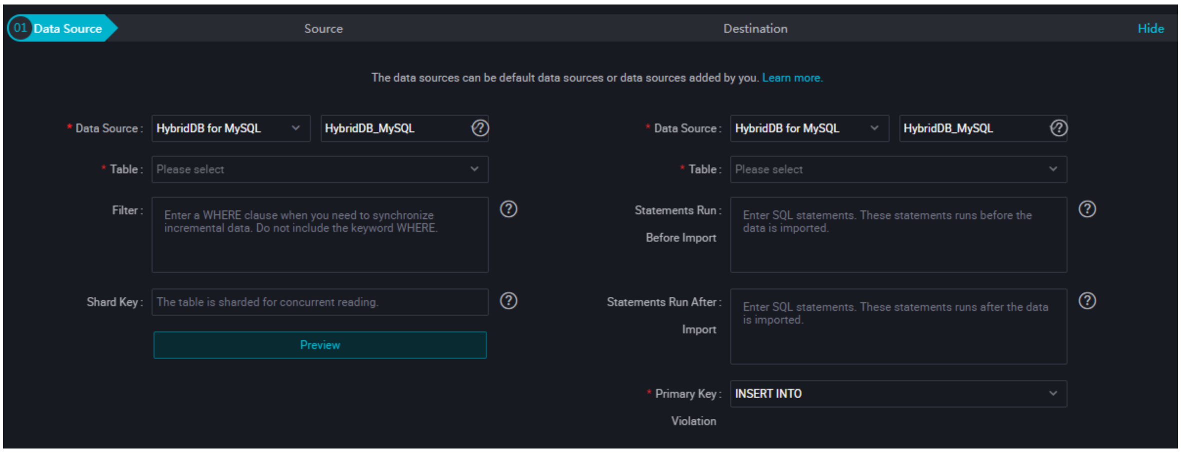
Parameter	Description	Required	Default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. The default value is [*], which indicates all the columns in the source table. • You can select specific columns to read. • The column order can be changed. This indicates that you can specify columns in an order different from the order specified by the schema of the source table. • Constants are supported. The column names must be arranged in compliance with the SQL syntax supported by HybridDB for MySQL Reader, such as `["id", "table","1","'mingya.wmy'","'null'", "to_char(a+1)","2.3","true"]` . ◦ id: a column name. ◦ table: the name of a column that contains reserved keywords. ◦ 1: an integer constant. ◦ 'mingya.wmy': a string constant, which is enclosed in single quotation marks ('). ◦ 'null': the string null. ◦ to_char(a+1): a function expression that is used to calculate the length of a string. ◦ 2.3: a floating-point constant. ◦ true: a Boolean value. • The column parameter must explicitly specify all the columns from which you want to read data. The parameter cannot be left empty.	Yes	No default value
splitPk	The field that is used for data sharding when HybridDB for MySQL Reader reads data. If you specify this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This improves data synchronization efficiency. • We recommend that you set the splitPk parameter to the name of the primary key column of the table. Data can be evenly distributed to different shards based on the primary key column, instead of being intensively distributed only to specific shards. • The splitPk parameter supports data sharding only for data of integer data types. If you set this parameter to a field of an unsupported data type, such as a string, floating point, or date data type, HybridDB for MySQL Reader ignores the setting of the splitPk parameter and uses a single thread to read data. • If the splitPk parameter is not provided or is left empty, HybridDB for MySQL Reader uses a single thread to read data.	No	No default value
where	The WHERE clause. For example, you can set this parameter to `gmt_create > $bizdate` to read the data that is generated on the current day. • You can use the WHERE clause to read incremental data. If the where parameter is not provided or is left empty, HybridDB for MySQL Reader reads all data. • Do not set the where parameter to limit 10, which does not conform to the constraints of HybridDB for MySQL on the SQL WHERE clause.	No	No default value

Parameter	Description	Required	Default value
querySql (advanced parameter, which is available only in the code editor)	The SQL statement that is used for refined data filtering. If you specify this parameter, HybridDB for MySQL Reader filters data based only on the value of this parameter. For example, if you want to join multiple tables for data synchronization, set this parameter to `["id","table","1",'''mingya.wmy''','''null''',"to_char(a+1)","2.3","true"]` . The priority of the querySql parameter is higher than those of the table, column, where, and splitPk parameters. If you specify the querySql parameter, HybridDB for MySQL Reader ignores the settings of the column, table, where, and splitPk parameters that you specified. The system parses information, such as the username and password, that is required by the data source specified by the datasource parameter from the querySql parameter.	No	No default value
singleOrMulti (available only for a sharded table)	Specifies that the source table is a sharded table. After you switch from the codeless UI to the code editor, the `"singleOrMulti":"multi"` configuration is automatically generated. However, if you use the code editor at the beginning, the configuration is not automatically generated, and you must manually add the configuration in the code editor. If you do not add the configuration, HybridDB for MySQL Reader reads data only from the first shard. The singleOrMulti parameter is used only at the frontend.	Yes	*multi*

Configure HybridDB for MySQL Reader by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



The data sources can be default data sources or data sources added by you. Learn more.

Source:

* Data Source : HybridDB for MySQL HybridDB_MySQL

* Table : Please select

Filter : Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.

Shard Key : The table is sharded for concurrent reading.

Destination:

* Data Source : HybridDB for MySQL HybridDB_MySQL

* Table : Please select

Statements Run Before Import : Enter SQL statements. These statements runs before the data is imported.

Statements Run After Import : Enter SQL statements. These statements runs after the data is imported.

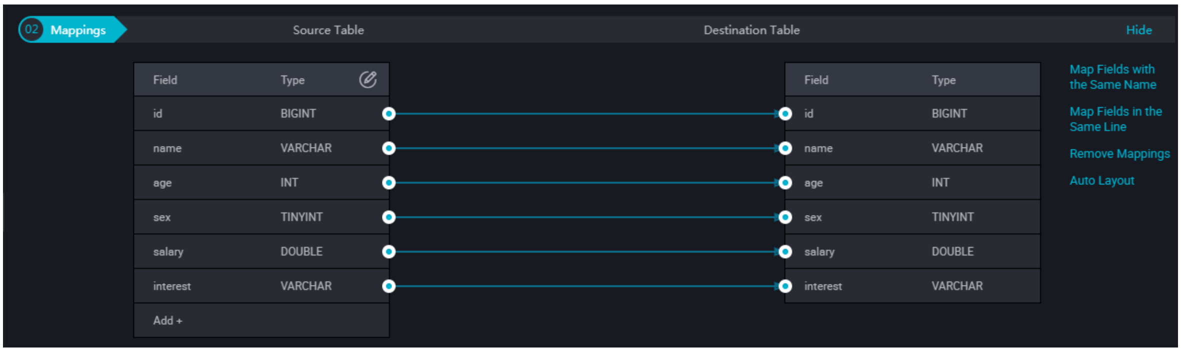
* Primary Key Violation : INSERT INTO

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table from which you want to read data. This parameter is equivalent to the table parameter that is described in the preceding section.
Filter	The condition that is used to filter the data you want to read. Filtering based on the LIMIT keyword is not supported. The SQL syntax is determined by the selected data source.

Parameter	Description
Shard Key	The shard key. You can use a column in the source table as the shard key. We recommend that you use the primary key column or an indexed column. Only integer columns are supported. If you specify this parameter, data sharding is performed based on the value of this parameter, and parallel threads can be used to read data. This improves data synchronization efficiency. Note The Shard Key parameter is displayed only after you select the data source for the synchronization node.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.

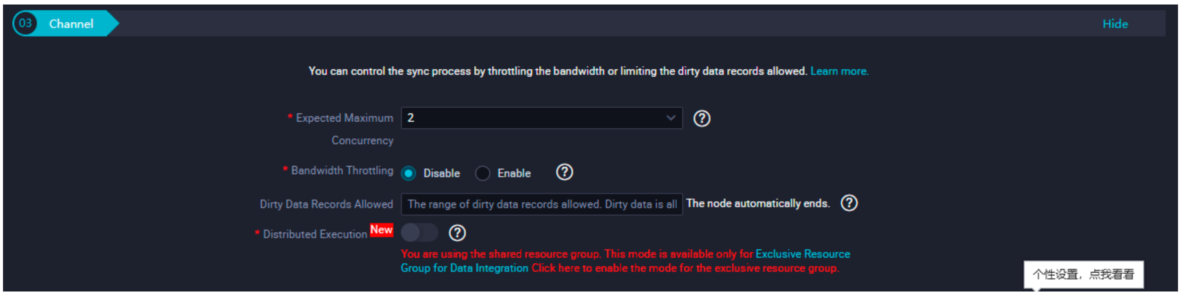


Field	Type
id	BIGINT
name	VARCHAR
age	INT
sex	TINYINT
salary	DOUBLE
interest	VARCHAR

Field	Type
id	BIGINT
name	VARCHAR
age	INT
sex	TINYINT
salary	DOUBLE
interest	VARCHAR

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click **Auto Layout**. Then, the system automatically sorts the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	Click Add to add a field. Take note of the following rules when you add a field: ○ You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ○ You can use scheduling parameters, such as ${bizdate}. ○ You can enter functions that are supported by relational databases, such as now() and count(1). ○ If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



03 Channel

Hide

You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure HybridDB for MySQL Reader by using the code editor

In the following code, a synchronization node is configured to read data from a table that is not sharded. For more information about the parameters, see the preceding parameter description.

```
{
    "type": "job",
    "steps": [
        {
            "parameter": {
                "datasource": "px_aliyun_hymysql",// The name of the data source.
                "column": [// The names of the columns from which you want to read data.
                    "id",
                    "name",
                    "sex",
                    "salary",
                    "age",
                    "pt"
                ],
                "where": "id=10001",// The WHERE clause.
                "splitPk": "id",// The shard key.
                "table": "person"// The name of the table from which you want to read data.
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "parameter": {}
    ],
    "version": "2.0",// The version number.
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {// The maximum number of dirty data records allowed.
            "record": ""
        },
        "speed": {
            "concurrent": 7,// The maximum number of parallel threads.
            "throttle": true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "mbps": 1 // The maximum transmission rate.
        }
    }
}
```

5.2.20. AnalyticDB for PostgreSQL Reader

This topic describes the data types and parameters that are supported by AnalyticDB for PostgreSQL Reader and how to configure AnalyticDB for PostgreSQL Reader by using the codeless user interface (UI) and code editor.

AnalyticDB for PostgreSQL Reader reads data from AnalyticDB for PostgreSQL.

AnalyticDB for PostgreSQL Reader connects to a remote AnalyticDB for PostgreSQL database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the statement on the database and returns data. Then, AnalyticDB for PostgreSQL Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

- AnalyticDB for PostgreSQL Reader generates the SQL statement based on the settings of the table, column, and where parameters and sends the generated statement to the remote database.
- If you specify the querySql parameter, AnalyticDB for PostgreSQL Reader sends the value of this parameter to the AnalyticDB for PostgreSQL database.

Data types

AnalyticDB for PostgreSQL Reader supports most AnalyticDB for PostgreSQL data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by AnalyticDB for PostgreSQL Reader.

Category	AnalyticDB for PostgreSQL data type
Integer	BIGINT, BIGSERIAL, INTEGER, SMALLINT, and SERIAL
Floating point	DOUBLE, PRECISION, MONEY, NUMERIC, and REAL
String	VARCHAR, CHAR, TEXT, BIT, and INET
Date and time	DATE, TIME, and TIMESTAMP
Boolean	BOOLEAN
Binary	BYTEA

Parameters

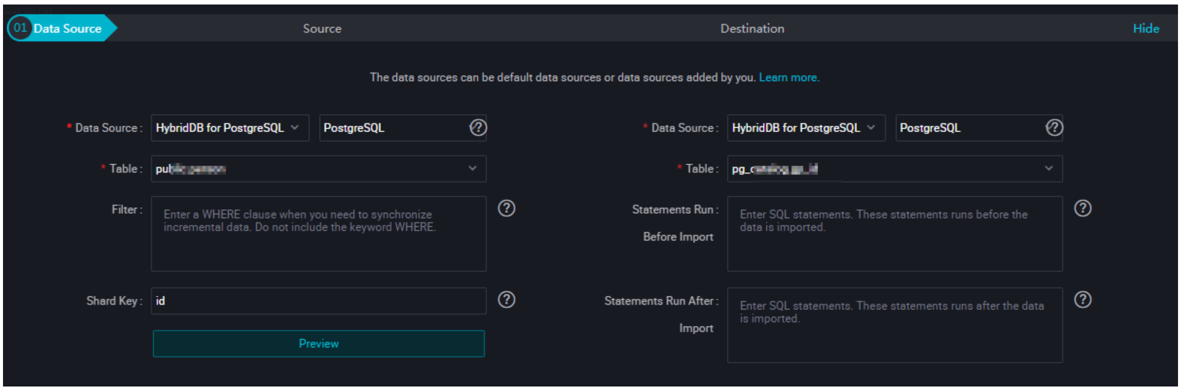
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table from which you want to read data.	Yes	No default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. The default value is [*], which indicates all the columns in the source table. • You can select specific columns to read. • The column order can be changed. This indicates that you can specify columns in an order different from the order specified by the schema of the source table. • Constants are supported. The column names must be arranged in compliance with the SQL syntax supported by AnalyticDB for PostgreSQL, such as `["id", "table","1","'mingya.wmy'","'null'", "to_char(a+1)","2.3","true"]` . ◦ id: a column name. ◦ table: the name of a column that contains reserved keywords. ◦ 1: an integer constant. ◦ 'mingya.wmy': a string constant, which is enclosed in single quotation marks ('). ◦ 'null': the string null. ◦ to_char(a+1): a function expression that is used to calculate the length of a string. ◦ 2.3: a floating-point constant. ◦ true: a Boolean value. • The column parameter must explicitly specify all the columns from which you want to read data. The parameter cannot be left empty.	Yes	No default value

Parameter	Description	Required	Default value
splitPk	The field that is used for data sharding when AnalyticDB for PostgreSQL Reader reads data. If you specify this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This improves data synchronization efficiency. • We recommend that you set the splitPk parameter to the name of the primary key column of the table. Data can be evenly distributed to different shards based on the primary key column, instead of being intensively distributed only to specific shards. • The splitPk parameter supports sharding only for data of integer data types. If you set this parameter to a field of an unsupported data type, such as a string, floating point, or date data type, AnalyticDB for PostgreSQL Reader ignores the setting of the splitPk parameter and uses a single thread to read data. • If the splitPk parameter is not provided or is left empty, AnalyticDB for PostgreSQL Reader uses a single thread to read data.	No	No default value
where	The WHERE clause. AnalyticDB for PostgreSQL Reader generates an SQL statement based on the settings of the table, column, and where parameters and uses the generated statement to read data. For example, when you perform a test, you can set the where parameter to `id>2 and sex=1` to read the data that is generated on the current day. • You can use the WHERE clause to read incremental data. • If the where parameter is not provided or is left empty, AnalyticDB for PostgreSQL Reader reads full data.	No	No default value
querySql (advanced parameter, which is available only in the code editor)	The SQL statement that is used for refined data filtering. If you specify this parameter, AnalyticDB for PostgreSQL Reader filters data based only on the value of this parameter. For example, if you want to join multiple tables for data synchronization, set this parameter to `select a,b from table_a join table_b on table_a.id = table_b.id` . If you specify this parameter, AnalyticDB for PostgreSQL Reader ignores the settings of the column, table, and where parameters.	No	No default value
fetchSize	The number of data records to read at a time. This parameter determines the number of interactions between Data Integration and the database and affects read efficiency. **Note** If you set this parameter to a value greater than 2048, an out of memory (OOM) error may occur during data synchronization.	No	*512*

Configure AnalyticDB for PostgreSQL Reader by using the codeless UI

1. Configure data sources.

Configure **Source** and **Target** for the synchronization node.



01 Data Source | Source | Destination | Hide

The data sources can be default data sources or data sources added by you. Learn more.

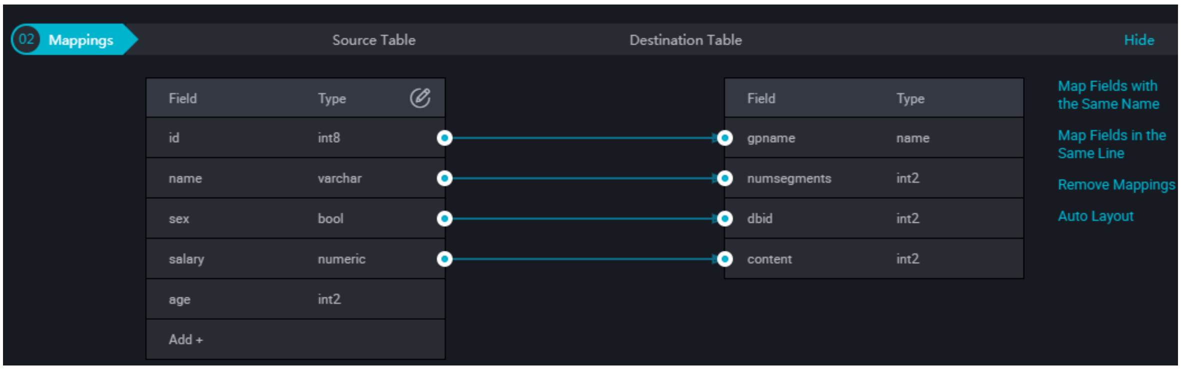
Source:
- * Data Source : HybridDB for PostgreSQL | PostgreSQL
- * Table : publ[redacted]
- Filter : Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.
- Shard Key : id
- Preview

Destination:
- * Data Source : HybridDB for PostgreSQL | PostgreSQL
- * Table : pg_[redacted]
- Statements Run Before Import : Enter SQL statements. These statements runs before the data is imported.
- Statements Run After Import : Enter SQL statements. These statements runs after the data is imported.

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table from which you want to read data. This parameter is equivalent to the table parameter that is described in the preceding section.
Filter	The condition that is used to filter the data you want to read. Filtering based on the LIMIT keyword is not supported. The SQL syntax is determined by the selected data source.
Shard Key	The shard key. You can use a column in the source table as the shard key. We recommend that you use the primary key column or an indexed column. Only integer columns are supported. If you specify this parameter, data sharding is performed based on the value of this parameter, and parallel threads can be used to read data. This improves data synchronization efficiency. **Note** The Shard Key parameter is displayed only after you select the data source for the synchronization node.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



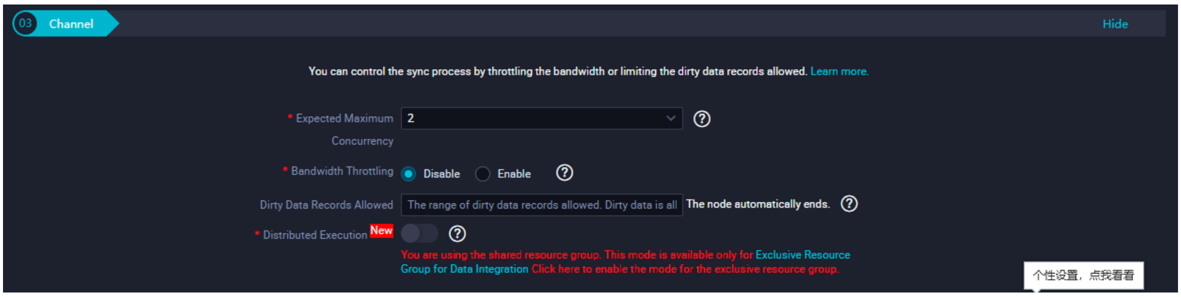
02 Mappings | Source Table | Destination Table | Hide

Field	Type	Field	Type
id	int8	gpname	name
name	varchar	numsegments	int2
sex	bool	dbid	int2
salary	numeric	content	int2
age	int2		
Add +			

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.

Operation	Description
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	Click Add to add a field. Take note of the following rules when you add a field: ○ You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ○ You can use scheduling parameters, such as ${bizdate}. ○ You can enter functions that are supported by relational databases, such as now() and count(1). ○ If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



03 Channel

Hide

You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure AnalyticDB for PostgreSQL Reader by using the code editor

```
{
    "type": "job",
    "steps": [
        {
            "parameter": {
                "datasource": "test_004",// The name of the data source.
                "column": [// The names of the columns from which you want to read data.
                    "id",
                    "name",
                    "sex",
                    "salary",
                    "age"
                ],
                "where": "id=1001",// The WHERE clause.
                "splitPk": "id",// The shard key.
                "table": "public.person"// The name of the table from which you want to read data.
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "parameter": {},
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",// The version number.
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {// The maximum number of dirty data records allowed.
            "record": ""
        },
        "speed": {
            "concurrent": 6,// The maximum number of parallel threads.
            "throttle": true,// Specifies whether to enable bandwidth throttling. The value false indicates tha
t bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps
parameter takes effect only when the throttle parameter is set to true.
           "mbps":"12"// The maximum transmission rate.
        }
    }
}
```

5.2.21. PolarDB Reader

This topic describes the data types and parameters that are supported by PolarDB Reader and how to configure PolarDB Reader by using the codeless user interface (UI) and code editor.

PolarDB Reader connects to a remote PolarDB database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the statement on the database and returns data. Then, PolarDB Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

PolarDB Reader can read tables and views. For tables, you can specify all or some of the columns in sequence, change the column order, specify constant fields, and configure functions supported by PolarDB, such as now().

Data types

The following table lists the data types that are supported by PolarDB Reader.

Category	PolarDB data type
Integer	INT, TINYINT, SMALLINT, MEDIUMINT, and BIGINT
Floating point	FLOAT, DOUBLE, and DECIMAL
String	VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT
Date and time	DATE, DATETIME, TIMESTAMP, TIME, and YEAR
Boolean	BIT and BOOLEAN
Binary	TINYBLOB, MEDIUMBLOB, BLOB, LONGBLOB, and VARBINARY

Note

- Data types that are not listed in the preceding table are not supported.
- PolarDB Reader processes TINYINT (1) as an integer data type.

Parameters

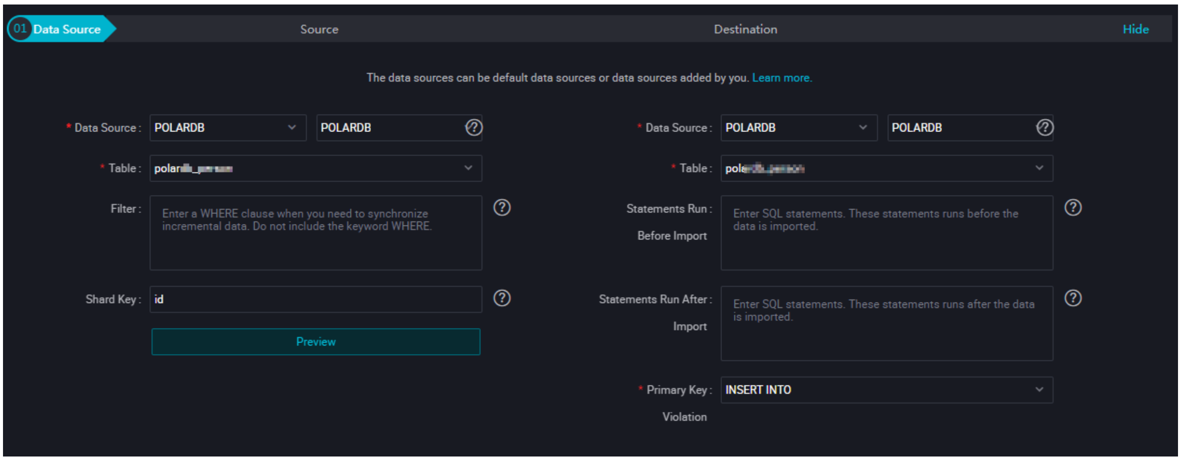
Operation	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table from which you want to read data.	Yes	No default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. The default value is [*], which indicates all the columns in the source table. • You can select specific columns to read. • The column order can be changed. This indicates that you can specify columns in an order different from the order specified by the schema of the source table. • Constants are supported. The column names must be arranged in compliance with the SQL syntax supported by PolarDB, such as `["id", "table","1","'mingya.wmy'","'null'", "to_char(a+1)","2.3","true"]` . ◦ id: a column name. ◦ table: the name of a column that contains reserved keywords. ◦ 1: an integer constant. ◦ 'mingya.wmy': a string constant, which is enclosed in single quotation marks ('). ◦ 'null': a string constant. ◦ to_char(a+1): a function expression that is used to calculate the length of a string. ◦ 2.3: a floating-point constant. ◦ true: a Boolean value. • The column parameter must explicitly specify all the columns from which you want to read data. This parameter cannot be left empty.	Yes	No default value

Operation	Description	Required	Default value
splitPk	The field that is used for data sharding when PolarDB Reader reads data. If you specify this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This way, data can be synchronized more efficiently. • We recommend that you set the splitPk parameter to the name of the primary key column of the table. Data can be evenly distributed to different shards based on the primary key column, but not intensively distributed only to specific shards. • The splitPk parameter supports sharding only for data of integer data types. If you set this parameter to a column of an unsupported data type, such as a string, floating point, or date data type, PolarDB Reader ignores the setting of the splitPk parameter and uses a single thread to read data. • If the splitPk parameter is not provided or is left empty, PolarDB Reader uses a single thread to read data.	No	No default value
where	The WHERE clause. For example, you can set this parameter to gmt_create > $bizdate in an actual business scenario to synchronize the data that is generated on the current day. • You can use the WHERE clause to read incremental data. If the where parameter is not provided or is left empty, PolarDB Reader reads all data. • Do not set the where parameter to limit 10. This value does not conform to the constraints of PolarDB on the SQL WHERE clause.	No	No default value
querySql (available only in the code editor)	The SQL statement that is used for refined data filtering. If you specify this parameter, Data Integration filters data based on the value of this parameter. For example, if you want to join multiple tables for data synchronization, you can set this parameter to `select a,b from table_a join table_b on table_a.id = table_b.id` . If you specify this parameter, PolarDB Reader ignores the settings of the column, table, and where parameters. The data source that is specified by the datasource parameter parses information, including the username and password, from this parameter.	No	No default value
singleOrMulti (suitable only for sharding)	Specifies whether to shard the source table. After you switch from the codeless UI to the code editor, the `"singleOrMulti":"multi"` configuration is automatically generated. However, if you use the code editor at the beginning, the configuration is not automatically generated, and you must manually specify this parameter. If you do not specify this parameter, PolarDB Reader can read data only from the first shard. **Note** The singleOrMulti parameter is used only by the frontend.	Yes	*multi*

Configure PolarDB Reader by using the codeless UI

1. Configure data sources.

Configure **Source** and **Target** for the synchronization node.



The data sources can be default data sources or data sources added by you. Learn more.

Source:

- * Data Source : POLARDB POLARDB
- * Table : [redacted]
- Filter : Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.
- Shard Key : id

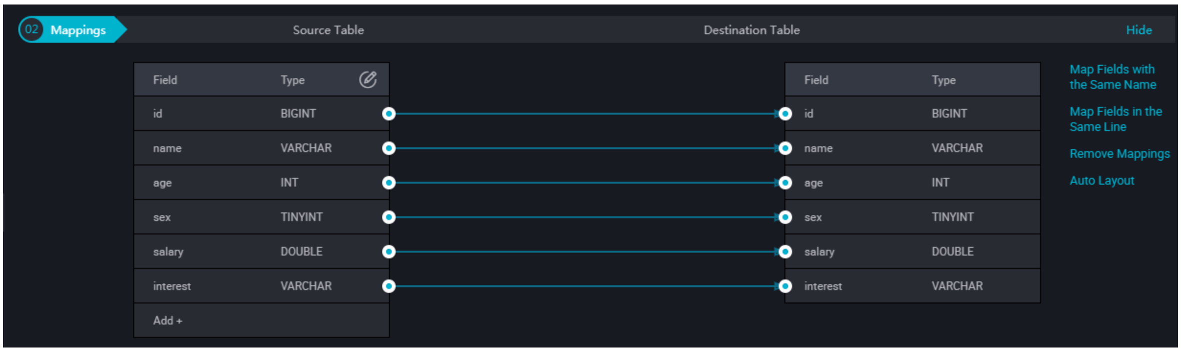
Destination:

- * Data Source : POLARDB POLARDB
- * Table : [redacted]
- Statements Run Before Import : Enter SQL statements. These statements runs before the data is imported.
- Statements Run After Import : Enter SQL statements. These statements runs after the data is imported.
- * Primary Key Violation : INSERT INTO

Operation	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table from which you want to read data. This parameter is equivalent to the table parameter that is described in the preceding section.
Data filter	The condition that is used to filter the data you want to read. Filtering based on the LIMIT keyword is not supported. The SQL syntax is determined by the selected data source.
Split pk	The shard key. You can use a column in the source table as the shard key. We recommend that you use the primary key column or an indexed column. Only integer columns are supported. If you specify this parameter, data sharding is performed based on the value of this parameter, and parallel threads can be used to read data. This improves data synchronization efficiency. **Note** The Split pk parameter is displayed only after you select the data source for the synchronization node.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



Source Table:

Field	Type
id	BIGINT
name	VARCHAR
age	INT
sex	TINYINT
salary	DOUBLE
interest	VARCHAR

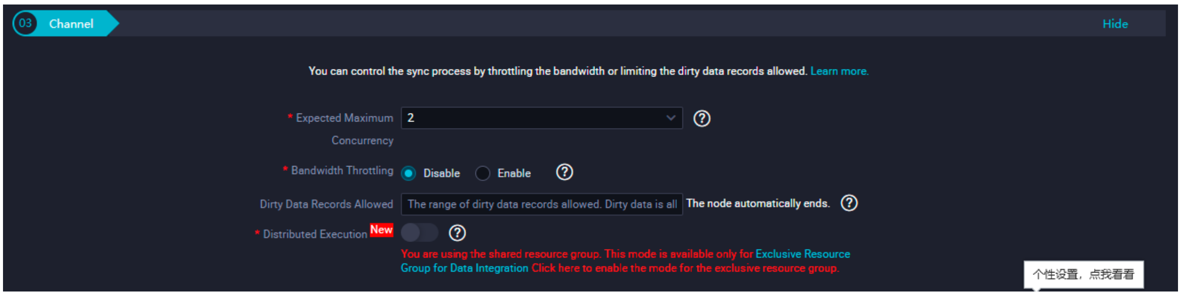
Destination Table:

Field	Type
id	BIGINT
name	VARCHAR
age	INT
sex	TINYINT
salary	DOUBLE
interest	VARCHAR

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.

Operation	Description
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Operation	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure PolarDB Reader by using the code editor

In the following code, a synchronization node is configured to read data from a table that is not sharded. For more information about the parameters, see the preceding parameter description.

```
{
    "type": "job",
    "steps": [
        {
            "parameter": {
                "datasource": "test_005",// The name of the data source.
                "column": [// The names of the columns from which you want to read data.
                    "id",
                    "name",
                    "age",
                    "sex",
                    "salary",
                    "interest"
                ],
                "where": "id=1001",// The WHERE clause.
                "splitPk": "id",// The shard key.
                "table": "PolarDB_person"// The name of the table from which you want to read data.
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "parameter": {}
    ],
    "version": "2.0",// The version number.
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {// The maximum number of dirty data records allowed.
            "record": ""
        },
        "speed": {
            "concurrent": 6,// The maximum number of parallel threads.
            "throttle":true// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
      "mbps":"12",// The maximum transmission rate.
        }
    }
}
```

5.2.22. Elasticsearch Reader

旧版本中文取消翻译

5.2.23. AnalyticDB for MySQL 2.0 Reader

This topic describes the data types and parameters that are supported by AnalyticDB for MySQL 2.0 Reader and how to configure AnalyticDB for MySQL 2.0 Reader by using the code editor.

AnalyticDB for MySQL 2.0 Reader reads data from tables stored in AnalyticDB for MySQL 2.0 databases. AnalyticDB for MySQL 2.0 Reader connects to a remote AnalyticDB for MySQL 2.0 database by using the Java Database Connectivity (JDBC) URL and executes SQL statements to read data from the AnalyticDB for MySQL 2.0 database in batches based on the recommended page size.

Data types

AnalyticDB for MySQL 2.0 data type	Data Integration data type	MaxCompute data type
BIGINT	LONG	BIGINT
TINYINT	LONG	INT
TIMESTAMP	DATE	DATETIME
VARCHAR	STRING	STRING
SMALLINT	LONG	INT
INT	LONG	INT
FLOAT	STRING	DOUBLE
DOUBLE	STRING	DOUBLE
DATE	DATE	DATETIME
TIME	DATE	DATETIME

Note AnalyticDB for MySQL 2.0 Reader does not support the multivalue type. If data of this type exists, AnalyticDB for MySQL 2.0 Reader unexpectedly exits.

Limits

In the current version, data export times out if a large amount of data is exported to a host with low configuration.

- If the mode parameter is set to Select, AnalyticDB for MySQL 2.0 Reader can read a maximum of 300,000 rows.
- If the mode parameter is set to ODPS, AnalyticDB for MySQL 2.0 Reader can read a maximum of 100 million rows.
- AnalyticDB for MySQL 2.0 limits the number of columns that can be read at a time to 50. If you want to cancel this limit, contact the administrator of the AnalyticDB for MySQL 2.0 database.
- The Java version must be 1.8 and later. `native2ascii LocalStrings.properties > LocalStrings_en_US.properties` must be used for converting the encoding format.

Parameters

Parameter	Description	Required	Default value
table	The name of the table to be exported.	Yes	No default value
column	The columns to be exported. If this parameter is not specified, all columns are exported.	No	*
limit	The maximum number of records that can be exported on one page.	No	No default value
where	The WHERE clause based on which data records are filtered. The string specified by this parameter, such as `where id < 100` , is added to SQL statements as the query condition.	No	No default value

Parameter	Description	Required	Default value
mode	The export type. Valid values: Select and ODPS. - Select: exports data on multiple pages based on the value specified for the limit parameter. - ODPS: uses MaxCompute dump to export data. You must have the permissions to access MaxCompute.	No	*Select*
odps.accessKey	The AccessKey secret of the Alibaba Cloud account that is used by AnalyticDB for MySQL 2.0 Reader to access MaxCompute. The account must have the Describe, Create, Select, Alter, Update, and Drop permissions. This parameter is required if the mode parameter is set to ODPS.	No	No default value
odps.accessId	The AccessKey ID of the Alibaba Cloud account that is used by AnalyticDB for MySQL 2.0 Reader to access MaxCompute. The account must have the Describe, Create, Select, Alter, Update, and Drop permissions. This parameter is required if the mode parameter is set to ODPS.	No	No default value
odps.odpsServer	The endpoint of the MaxCompute API. This parameter is required if the mode parameter is set to ODPS.	No	No default value
odps.tunnelServer	The endpoint of MaxCompute Tunnel. This parameter is required if the mode parameter is set to ODPS.	No	No default value
odps.project	The name of the MaxCompute project. This parameter is required if the mode parameter is set to ODPS.	No	No default value
odps.accountType	The type of the account that is used to access MaxCompute. This parameter is required if the mode parameter is set to ODPS.	No	*aliyun*

Configure AnalyticDB for MySQL 2.0 Reader by using the code editor

```
{
    "type": "job",
    "steps": [
        {
            "stepType": "ads",
            "parameter": {
                "datasource": "ads_demo",
                "table": "th_test",
                "column": [
                    "id",
                    "testtinyint",
                    "testbigint",
                    "testdate",
                    "testtime",
                    "testtimestamp",
                    "testvarchar",
                    "testdouble",
                    "testfloat"
                ],
                "odps": {
                    "accessId": "<yourAccessKeyId>",
                    "accessKey": "<yourAccessKeySecret>",
                    "account": "*********@aliyun.com",
                    "odpsServer": " http://service.cn.maxcompute.aliyun-inc.com/api",
                    "tunnelServer": "http://dt.cn-shanghai.maxcompute.aliyun-inc.com",
                    "accountType": "aliyun",
                    "project": "odps_test"
                },
                "mode": "ODPS"
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "stream",
            "parameter": {},
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": ""
        },
        "speed": {
            "concurrent": 2,
            "throttle": true,// Specifies whether to enable bandwidth throttling. A value of false indicates th
at bandwidth throttling is disabled, and a value of true indicates that bandwidth throttling is enabled. The mb
ps parameter takes effect only if the throttle parameter is set to true.
                  "mbps":"12"// The maximum transmission rate.
        }
    }
}
```

5.2.24. Kafka Reader

Kafka Reader reads data from Kafka by using Kafka SDK for Java.

Background information

Apache Kafka is a fast, scalable, high-throughput, and distributed messaging system that supports fault tolerance. This system is used to publish and subscribe to messages. Kafka provides built-in partitions, supports data replicas, and can be used to process a large number of messages.

> **Notice**
>
> - Kafka Reader can be used to read data from Message Queue for Apache Kafka data sources and self-managed Kafka data sources. However, the versions of self-managed Kafka data sources must range from 0.10.2 to 2.2.x.
> - Self-managed Kafka data sources whose versions are earlier than 0.10.2 do not support the query of offsets of partition data and do not support timestamps. Data synchronization cannot be performed.
> - Kafka Reader supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.

How it works

Kafka Reader reads data from Kafka by using Kafka SDK for Java of the following version:

```
<dependency>
   <groupId>org.apache.kafka</groupId>
   <artifactId>kafka-clients</artifactId>
   <version>2.0.0</version>
</dependency>
```

The following methods of Kafka SDK for Java are used by Kafka Reader. For more information, see Kafka documentation.

- Use a Kafka consumer as a client that consumes messages.

  ```
  org.apache.kafka.clients.consumer.KafkaConsumer<K,V>
  ```

- Query offsets based on the UNIX timestamp.

  ```
  Map<TopicPartition,OffsetAndTimestamp> offsetsForTimes(Map<TopicPartition,Long> timestampsToSearch)
  ```

- Seek to the start offset.

  ```
  public void seekToBeginning(Collection<TopicPartition> partitions)
  ```

- Seek to the end offset.

  ```
  public void seekToEnd(Collection<TopicPartition> partitions)
  ```

- Seek to a specific offset.

  ```
  public void seek(TopicPartition partition,long offset)
  ```

- Poll for data from the server.

  ```
  public ConsumerRecords<K,V> poll(final Duration timeout)
  ```

> **Note** Kafka Reader automatically commits offsets when it consumes data.

Parameters

Parameter	Description	Required

Parameter	Description	Required
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes
server	The address of a Kafka broker in your Kafka cluster. Specify the address in the following format: IP address:Port number. You can specify one or more broker addresses in the server parameter. You must make sure that DataWorks can use the specified addresses to access the related brokers.	Yes
topic	The name of the Kafka topic from which you want to read data. Kafka maintains feeds of messages in topics.	Yes

column	The names of the columns from which you want to read data. Constant columns, data columns, and property columns are supported.
	- Constant column: a column whose name is enclosed in single quotation marks ('). Example: `["'abc'", "'123'"]` .
	- Data column:
	  ○ If your data is in the JSON format, you can obtain JSON properties. Example: `["event_id"]` .
	  ○ If your data is in the JSON format, you can obtain the properties of nested objects in the data. Example: `["tag.desc"]` .
	- Property column:
	  ○ __key__: the key of a Kafka record.
	  ○ __value__: the complete content of a Kafka record.
	  ○ __partition__: the partition where a Kafka record resides.
	  ○ __headers__: the header of a Kafka record.
	  ○ __offset__: the offset of a Kafka record.
	  ○ __timestamp__: the timestamp of a Kafka record.
	The following code provides a configuration example of the column parameter:
	`"column": [`
	`    "__key__",`
	`    "__value__",`
	`    "__partition__",`
	`    "__offset__",`
	`    "__timestamp__",`
	`    "'123'",`
	`    "event_id",`
	`    "tag.desc"`
	`    ]`
		Yes

keyType	The data type of the key in the Kafka topic. Valid values: BYTEARRAY, DOUBLE, FLOAT, INTEGER, LONG, and SHORT.	Yes
valueType	The data type of the value in the Kafka topic. Valid values: BYTEARRAY, DOUBLE, FLOAT, INTEGER, LONG, and SHORT.	Yes

Parameter	Description	Required
beginDateTime	The start time of data consumption. This parameter specifies the left boundary of a left-closed, right-open interval. Specify the time in the yyyyMMddHHmmss format. This parameter can be used together with the **scheduling time parameters** in DataWorks. For more information, see Overview of scheduling parameters. **Note** This parameter is supported by Kafka 0.10.2 and later.	You must configure either the beginDateTime or beginOffset parameter. **Note** The beginDateTime and endDateTime parameters must be used in pairs.
endDateTime	The end time of data consumption. This parameter specifies the right boundary of a left-closed, right-open interval. Specify the time in the yyyyMMddHHmmss format. This parameter can be used together with the **scheduling time parameters** in DataWorks. For more information, see Overview of scheduling parameters. **Note** This parameter is supported by Kafka 0.10.2 and later.	You must configure either the endDateTime or endOffset parameter. **Note** The beginDateTime and endDateTime parameters must be used in pairs.
beginOffset	The offset from which data consumption starts. The following formats are supported: - Numeric string: Data consumption starts from the specified offset, such as 15553274. - seekToBeginning: Data is consumed from the start offset. - seekToLast: Data is consumed from the last offset. - seekToEnd: Data is consumed from the end offset. In this case, null may be read.	You must configure either the beginOffset or beginDateTime parameter.
endOffset	The offset at which data consumption ends.	You must configure either the endOffset or endDateTime parameter.
skipExceedRecord	The Kafka consumer uses `public ConsumerRecords<K, V> poll(final Duration timeout)` to poll for data. However, the data obtained in a poll may be beyond the boundary that is specified by the endOffset or endDateTime parameter. The skipExceedRecord parameter specifies whether to poll for the excess data. Kafka Reader automatically commits offsets for data consumption. Therefore, we recommend that you configure the skipExceedRecord parameter based on the following instructions: - If the data source from which you want to read data is of a version earlier than Kafka 0.10.2, set the skipExceedRecord parameter to false. - If the data source from which you want to read data is of Kafka 0.10.2 or later, set the skipExceedRecord parameter to true.	No. Default value: *false*.
partition	If a Kafka topic contains multiple partitions, Kafka Reader reads data in a specific offset interval from the partitions. If you want Kafka Reader to read data from a specific partition, you can use the partition parameter to specify the partition.	No. This parameter does not have a default value.
kafkaConfig	The extended parameters that are specified when you create the Kafka consumer, such as bootstrap.servers, auto.commit.interval.ms, and session.timeout.ms. You can configure the parameters in kafkaConfig to manage the data consumption of the Kafka consumer.	No
encoding	If the keyType or valueType parameter is set to STRING, strings are parsed based on the value of the encoding parameter.	No. Default value: UTF-8.

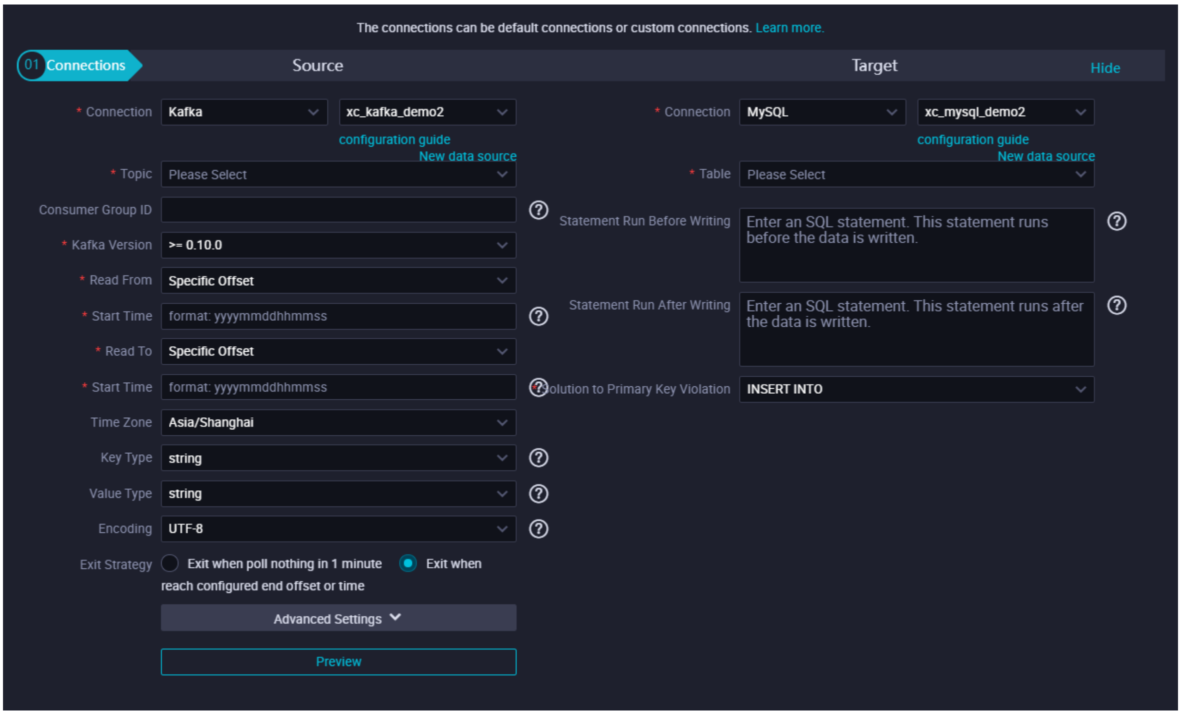
Parameter	Description	Required
waitTIme	The maximum duration during which the Kafka consumer waits to poll for data from Kafka each time. Unit: seconds.	No. Default value: 60.
stopWhenPollEmpty	You can set this parameter to true or false. If you set this parameter to true, the Kafka consumer may poll for null from Kafka because all data in the Kafka topic is polled or an error occurs on the network or Kafka cluster. If the Kafka consumer polls for null, the related synchronization node immediately stops. If you set this parameter to false and the Kafka consumer polls for null, the Kafka consumer attempts to poll for data until the data is obtained.	No. Default value: true.

The following table describes the parameters in kafkaConfig.

Parameter	Description
fetch.min.bytes	The minimum number of bytes that the consumer can obtain from the broker. The broker returns data to the consumer only after the number of bytes reaches the specified value.
fetch.max.wait.ms	The maximum duration during which the consumer waits for data from the broker. Default value: 500. Unit: milliseconds. The broker returns data to the consumer when one of the conditions that are specified by the fetch.min.bytes and fetch.max.wait.ms parameters is met.
max.partition.fetch.bytes	The maximum number of bytes in each partition that the broker returns to the consumer. Default value: 1. Unit: MB.
session.timeout.ms	The timeout period of a connection session between the consumer and the broker. If this limit is exceeded, the consumer can no longer receive data from the broker. Default value: 30. Unit: seconds.
auto.offset.reset	The handling method that is used when no offset is available or the offset is invalid. This occurs if the consumer times out or the record with the specified offset expires and is deleted. The default value of this parameter is latest, which indicates that the consumer reads data from the latest record. You can set this parameter to earliest, which indicates that the consumer reads data from the start offset.
max.poll.records	The number of Kafka records that can be returned for a single poll.
key.deserializer	The method that is used to deserialize the Kafka record key, such as org.apache.kafka.common.serialization.StringDeserializer.
value.deserializer	The method that is used to deserialize the Kafka record value, such as org.apache.kafka.common.serialization.StringDeserializer.
ssl.truststore.location	The path of the Secure Sockets Layer (SSL) root certificate.
ssl.truststore.password	The password of the truststore in the SSL root certificate. If you use Message Queue for Apache Kafka, set this parameter to KafkaOnsClient.
security.protocol	The access protocol. Set this parameter to SASL_SSL.
sasl.mechanism	The simple authentication and security layer (SASL) authentication mode. If you use Message Queue for Apache Kafka, set this parameter to PLAIN.
java.security.auth.login.config	The path of the SASL authentication file.

Configure Kafka Reader by using the codeless UI

1. Configure data sources. Configure **Source** and **Target** for the synchronization node.



The connections can be default connections or custom connections. Learn more.

01 Connections | Source | Target | Hide

Source:

- * Connection: Kafka | xc_kafka_demo2 (configuration guide, New data source)
- * Topic: Please Select
- Consumer Group ID:
- * Kafka Version: >= 0.10.0
- * Read From: Specific Offset
- * Start Time: format: yyyymmddhhmmss
- * Read To: Specific Offset
- * Start Time: format: yyyymmddhhmmss
- Time Zone: Asia/Shanghai
- Key Type: string
- Value Type: string
- Encoding: UTF-8
- Exit Strategy: ○ Exit when poll nothing in 1 minute ● Exit when reach configured end offset or time

Advanced Settings

Preview

Target:

- * Connection: MySQL | xc_mysql_demo2 (configuration guide, New data source)
- * Table: Please Select
- Statement Run Before Writing: Enter an SQL statement. This statement runs before the data is written.
- Statement Run After Writing: Enter an SQL statement. This statement runs after the data is written.
- Solution to Primary Key Violation: INSERT INTO

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Topic	The name of the Kafka topic from which you want to read data. This parameter is equivalent to the topic parameter that is described in the preceding section.
Consumer Group ID	The ID of the consumer group to which the Kafka consumer that you want to initialize belongs. This parameter is equivalent to the group.id field in the JSON code of kafkaConfig that is described in the preceding section. If you want a data synchronization node in Data Integration to consume data from the correct offset, you must set this parameter to a value that is unique to the data synchronization node. If you do not specify a value for this parameter, a random string that starts with datax_ is automatically generated for group.id each time data is synchronized.
Kafka Version	The Kafka version. Valid values: >=0.10.2 and <0.10.2. The value of this parameter affects the options available for the Read From and Read To parameters.

Parameter	Description
Read From	The offset from which data consumption starts. This parameter is equivalent to the beginOffset parameter that is described in the preceding section. ○ Specific Offset: A UNIX timestamp is automatically generated and used as the data timestamp of each Kafka record that you want to write. The data synchronization node converts the time in the yyyyMMddHHmmss format into a UNIX timestamp, and reads data from the earliest offset in the partitions whose data timestamp is later than or equal to the UNIX timestamp in a Kafka topic. For example, you can set Specific Offset to 20210125000000, which is the same as the value of beginDateTime. ○ Earliest Offset: Kafka Reader starts to read data from the first Kafka record that remains in each partition of a Kafka topic. ○ Consumer Group Current Offset: Kafka Reader starts to read data from the offset that is saved based on the consumer group ID that you specify on the configuration page of the data synchronization node. In most cases, the offset is the position at which the previous data read stops. We recommend that you configure only one data synchronization node in Data Integration for the consumer that uses this consumer group ID. If multiple synchronization nodes in Data Integration use the same consumer group ID, data loss may occur. If you want to use the current offset of the consumer group as the start offset, you must specify a consumer group ID. If you do not specify the consumer group ID, a random consumer group ID is generated for the data synchronization node, and Kafka Reader starts to read data from the earliest offset in a partition because no offset is saved based on the generated consumer group ID. **Note** If you set Read From to Earliest Offset or Consumer Group Current Offset, the value of the beginOffset parameter is replaced with the value of the Earliest Offset or Consumer Group Current Offset parameter. If you set Read From to Specific Offset, the value of the beginOffset parameter is not replaced with the value of the Specific Offset parameter after node configurations are generated. Instead, the beginDateTime parameter value that is replaced with the value of the Start Time parameter determines the offset from which data consumption starts.
Start Time	The start time of data consumption. This parameter is equivalent to the beginDateTime parameter that is described in the preceding section. If you set Read From to Specific Offset, the value of this parameter is a time string in the yyyyMMddHHmmss format. This parameter specifies the left boundary of a left-closed, right-open interval, such as 20210513000000. You can use this parameter with the scheduling time parameters in DataWorks. For more information, see Configure scheduling parameters.
Read To	The offset at which data consumption ends. This parameter is equivalent to the endOffset parameter that is described in the preceding section. **Note** If you set Read To to Latest Offset, the value of the endOffset parameter is replaced with the value of the Latest Offset parameter. If you set Read To to Specific Offset, the value of the endOffset parameter is not replaced with the value of the Specific Offset parameter after node configurations are generated. Instead, the endDateTime parameter value that is replaced by using an end time table determines the offset at which data consumption ends.
End Time	The end time of data consumption. This parameter is equivalent to the endDateTime parameter that is described in the preceding section. If you set Read To to Specific Offset, the value of this parameter is a time string in the yyyyMMddHHmmss format. This parameter specifies the right boundary of a left-closed, right-open interval, such as 20210514000000. You can use this parameter with the scheduling time parameters in DataWorks. For more information, see Configure scheduling parameters.
Time Zone	The time zone that corresponds to the interval if you set Read From to Specific Offset and set Read To to Specific Offset.

Parameter	Description
Key Type	The data type of the key in the Kafka topic. This parameter is equivalent to the keyType parameter that is described in the preceding section. The value of this parameter determines the value of the key.deserializer parameter when the Kafka consumer is initialized. Valid values: STRING, BYTEARRAY, DOUBLE, FLOAT, INTEGER, LONG, and SHORT.
Value Type	The data type of the value in the Kafka topic. This parameter is equivalent to the valueType parameter that is described in the preceding section. The value of this parameter determines the value of the value.deserializer parameter when the Kafka consumer is initialized. Valid values: STRING, BYTEARRAY, DOUBLE, FLOAT, INTEGER, LONG, and SHORT.
Encoding	The encoding method that is used to read data when the Key Type or Value Type parameter is set to STRING. This parameter is equivalent to the encoding parameter that is described in the preceding section.
Exit Strategy	○ If you set this parameter to Exit when poll nothing in 1 minute, the Kafka consumer may poll for null from Kafka within 1 minute because all data in the Kafka topic is polled or an error occurs on the network or Kafka cluster. If the Kafka consumer polls for null, the synchronization node immediately stops. If you set this parameter to Exit when reach configured end offset or time and the Kafka consumer polls for null, the Kafka consumer attempts to poll for data until the data is obtained. ○ If you set this parameter to **Exit when reach configured end offset or time**, the synchronization node stops when the data timestamp or the configured end offset of the Kafka record that the synchronization node reads is reached. Otherwise, the Kafka consumer attempts to poll for data until the data is obtained.
Auto Offset Reset	The method that is used when no offset is available or the offset is invalid. This parameter is equivalent to the auto.offset.reset field in the JSON code of kafkaConfig that is described in the preceding section. We recommend that you set this parameter to none to ensure that all data is read.
Read Batch Size	The minimum number of bytes that the consumer can obtain from the broker. This parameter is equivalent to the fetch.min.bytes field in the JSON code of kafkaConfig that is described in the preceding section.
Read Wait Time	The maximum duration during which the consumer waits for data from the broker. This parameter is equivalent to the fetch.max.wait.ms field in the JSON code of kafkaConfig that is described in the preceding section.
Read Timeout	The timeout period of a connection session between the consumer and the broker. If this limit is exceeded, the consumer can no longer receive data from the broker. This parameter is equivalent to the session.timeout.ms field in the JSON code of kafkaConfig that is described in the preceding section.

2. Configure field mapping. This operation is equivalent to specifying a value for the column parameter that is described in the preceding section. Fields in the source on the left side have a one-to-one mapping with fields in the destination on the right side. You can click **Add** to add a field. To remove a field, move the pointer over the field and click the **Remove** icon.

   The following table describes the Kafka columns from which you want to read data. The names of the columns start with two underscores (_).

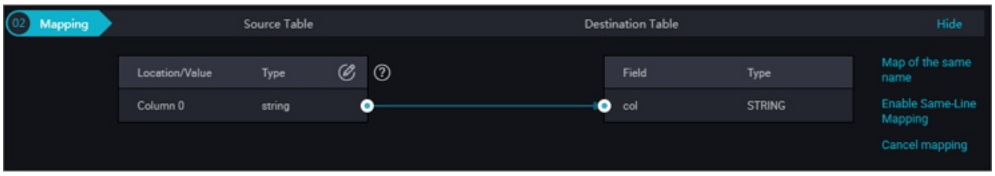
Source table field	Description
__key__	The key of a Kafka record.
__value__	The value of a Kafka record.
__partition__	The partition in which a Kafka record resides. The field value is an integer that is greater than or equal to 0.
__headers__	The JSON string that is obtained after you serialize the header of a Kafka record.

Source table field	Description
__offset__	The offset of a Kafka record. The field value is an integer that is greater than or equal to 0.
__timestamp__	The timestamp of a Kafka record. Unit: milliseconds.

You can also read data from other columns. Kafka Reader parses Kafka records as JSON strings and reads data from the source table field you specify, and a writer writes the data to the destination. In the following example, the source table field is name and Kafka Readers reads bob from this field and a writer writes bob to the destination table.

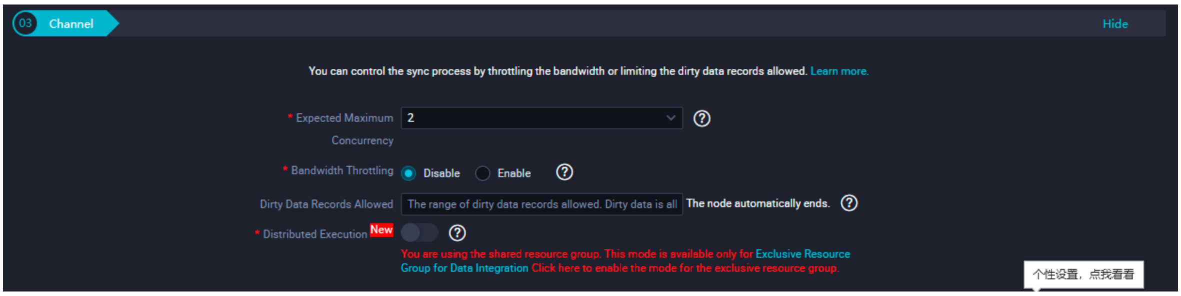
```
{
  "data": {
    "name": "bob",
    "age": 35
  }
}
```

The following figure shows the field mappings between the source table and the destination table.



Parameter	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	○ Click Add to add a field. Take note of the following rules when you add a field: You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ○ You can use scheduling parameters, such as ${bizdate}. ○ If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to avoid heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure Kafka Reader by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following JSON code, a synchronization node is configured to read data from a Kafka topic:

```
{
    "type": "job",
    "steps": [
        {
            "stepType": "kafka",
            "parameter": {
                "server": "host:9093",
                "column": [
                    "__key__",
                    "__value__",
                    "__partition__",
                    "__offset__",
                    "__timestamp__",
                    "'123'",
                    "event_id",
                    "tag.desc"
                ],
                "kafkaConfig": {
                    "group.id": "demo_test"
                },
                "topic": "topicName",
                "keyType": "ByteArray",
                "valueType": "ByteArray",
                "beginDateTime": "20190416000000",
                "endDateTime": "20190416000006",
                "skipExceedRecord": "false"
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "stream",
            "parameter": {
                "print": false,
                "fieldDelimiter": ","
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "throttle": true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent": 1,// The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    }
}
```

Use SASL authentication

If you want to use SASL or SSL for authentication, configure the SASL or SSL authentication mode when you configure a Kafka data source. For more information, see Add a Kafka data source.

5.2.25. MaxCompute Reader

This topic describes the data types and parameters that are supported by MaxCompute Reader and how to configure MaxCompute Reader by using the codeless user interface (UI) and code editor.

Background information

MaxCompute Reader reads data from MaxCompute. For more information about MaxCompute, see What is MaxCompute?

MaxCompute Reader uses Tunnel commands to read data from MaxCompute based on the information that you specify, such as the source project, table, partitions, and fields. For more information about common Tunnel commands, see Tunnel commands.

MaxCompute Reader cannot read views. It can read only partitioned and non-partitioned tables. DataWorks cannot map the fields in partitioned MaxCompute tables. If you want to read data from a partitioned MaxCompute table, you must specify each desired partition when you configure MaxCompute Reader. For example, if you want to read data from the partition pt=1,ds=hangzhou in the t0 table, you must specify pt=1,ds=hangzhou when you configure MaxCompute Reader. In addition, you can select some or all of the table fields, change the order in which the fields are arranged, or add constant fields and partition key columns. Partition key columns are not table fields.

Note

- MaxCompute Reader cannot filter data. If you want MaxCompute Reader to read only specific data during data synchronization, you must create a table, write the data to the table, and then enable MaxCompute Reader to read the data from the table.
- MaxCompute Reader cannot read data from external tables.

Data types

The following table lists the data types that are supported by MaxCompute Reader.

Category	Data Integration data type	MaxCompute data type
Integer	LONG	BIGINT, INT, TINYINT, and SMALLINT
Boolean	BOOLEAN	BOOLEAN
Date and time	DATE	DATETIME, TIMESTAMP, and DATE
Floating point	DOUBLE	FLOAT, DOUBLE, and DECIMAL
Binary	BYTES	BINARY
Complex	STRING	ARRAY, MAP, and STRUCT

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table from which you want to read data. The name is not case-sensitive.	Yes	No default value

Parameter	Description	Required	Default value
partition	The partitions from which you want to read data. - You can use Linux Shell wildcards to specify the partitions. An asterisk (`*`) indicates multiple numbers of characters, and a question mark (`?`) indicates a single character. - The partitions that you specify must exist in the source table. Otherwise, the system reports an error for the synchronization node. If you want the synchronization node to be successfully run even if the partitions that you specify do not exist in the source table, use the code editor to modify the code of the node. In addition, you must add `"successOnNoPartition": true` to the configuration of MaxCompute Reader. For example, the partitioned table *test* contains four partitions: *pt=1,ds=hangzhou*, *pt=1,ds=shanghai*, *pt=2,ds=hangzhou*, and *pt=2,ds=beijing*. In this case, you can set the partition parameter based on the following instructions: - To read data from the partition *pt=1,ds=hangzhou*, specify `"partition":"pt=1,ds=`*hangzhou*`"` . - To read data from all the ds partitions in the *pt=1* partition, specify `"partition":"pt=1,ds=*"` . - To read data from all the partitions in the *test* table, specify `"partition":"pt=*,ds=*"` . You can also specify other conditions to read data from partitions based on your business requirements. - To read data from the partition that stores the largest amount of data, add `/*query*/ ds=(select MAX(ds) from DataXODPSReaderPPR)` to the configuration of MaxCompute Reader. - To filter data by specifying filter conditions, add `/*query*/ pt+Expression` to the configuration of MaxCompute Reader. For example, `/*query*/ pt>=20170101 and pt<20170110` indicates that you want to read the data that is generated from January 1, 2017 to January 9, 2017 from all the pt partitions in the table test. **Note** MaxCompute Reader processes the content after `/*query*/` as a WHERE clause.	Required only for partitioned tables	No default value

Parameter	Description	Required	Default value
column	The names of the columns from which you want to read data. For example, the test table contains the id, name, and age columns. • To read the data in the columns in sequence, specify `"column":["id","name","age"]` or `"column":["*"]`. Note We recommend that you do not use "column":["*"]. If you specify "column":["*"], MaxCompute Reader reads data from all the columns in a source table in sequence. If the column order, data type, or number of columns is changed in the source table, the columns in the source and destination tables may be inconsistent. As a result, the data synchronization may fail, or the data synchronization results do not meet your expectation. • To read the data in the name and id columns in sequence, specify `"column":["name","id"]`. • You can add constant fields to the source table to establish mappings between the source table columns and destination table columns. In this case, when you specify the column parameter, you must enclose each constant field in single quotation marks (`'`). For example, if you add the constant field 1988-08-08 08:08:08 to the source table and want to read data from the age, name, 1988-08-08 08:08:08, and id columns in sequence, specify `"column":["age","name","'1988-08-08 08:08:08'","id"]`. The single quotation marks (`'`) are used to identify constant columns. When MaxCompute Reader reads data from the source table, the constant column values that are read by MaxCompute Reader exclude the single quotation marks (`'`). Note ◦ MaxCompute Reader does not use SELECT statements to read data. Therefore, you cannot specify function fields. ◦ The column parameter must explicitly specify all the columns from which you want to read data. The parameter cannot be left empty.	Yes	No default value

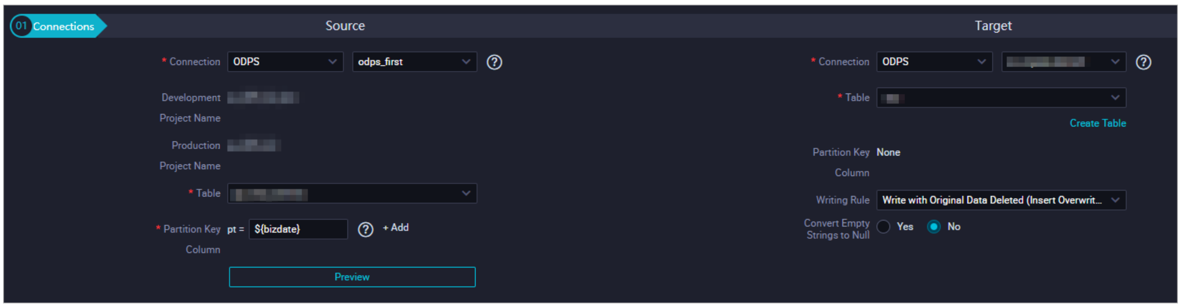
Configure MaxCompute Reader by using the codeless UI

Create a synchronization node and configure the node. For more information, see Configure a synchronization node by using the codeless UI.

Perform the following steps on the configuration tab of the synchronization node:

1. Configure data sources.

Configure **Source** and **Target** for the synchronization node.



Source

- * Connection: ODPS, odps_first
- Development Project Name: [redacted]
- Production Project Name: [redacted]
- * Table: [redacted]
- * Partition Key Column: pt = ${bizdate}

Target

- * Connection: ODPS, [redacted]
- * Table: [redacted]
- Partition Key Column: None
- Writing Rule: Write with Original Data Deleted (Insert Overwrit...
- Convert Empty Strings to Null: ○ Yes ● No

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Development Project Name	The name of the project in the development environment. You cannot change the value. Note This parameter is displayed only if the workspace is in standard mode.
Production Project Name	The name of the project in the production environment. You cannot change the value.
Table	The name of the table from which you want to read data. This parameter is equivalent to the table parameter that is described in the preceding section.
Partition Key Column	If your daily incremental data is stored in the partitions of a specific date, you can specify the partition information to synchronize the daily incremental data. For example, set pt to ${bizdate}. Note DataWorks cannot map the fields in partitioned MaxCompute tables. If you want to read data from a partitioned MaxCompute table, you must specify each desired partition when you configure MaxCompute Reader.

Note In the code editor, if you want to synchronize all the columns in the source table, specify `"column": ["*"]`. You can directly specify the partitions from which you want to read data. You can also use wildcards to specify the partitions.

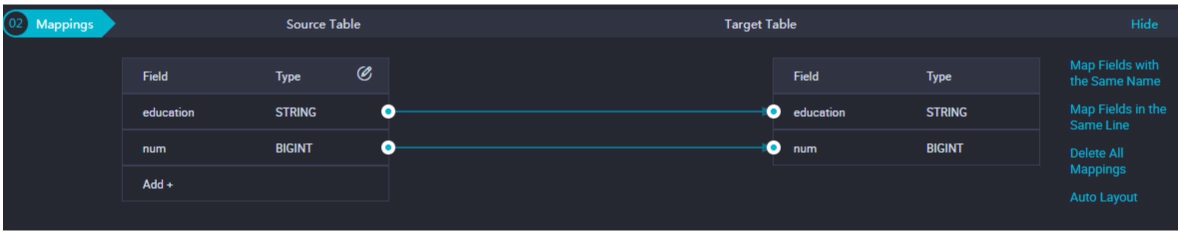
- `"partition":"pt=20140501/ds=*"` indicates all the ds partitions in the pt=20140501 partition.
- `"partition":"pt=top?"` indicates partitions pt=top and pt=to.

You can specify the partitions from which you want to synchronize data. For example, if you want to synchronize data from the partition pt=${bdp.system.bizdate} in a MaxCompute table, you can add the pt column to the source table in the Mappings section. If the value of Type for pt is Unidentified, you can ignore the value and proceed to the next step.

- To synchronize data in all partitions, specify pt=* for the Partition Key Column parameter.
- To synchronize data in specific partitions, specify the required dates.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



Source Table

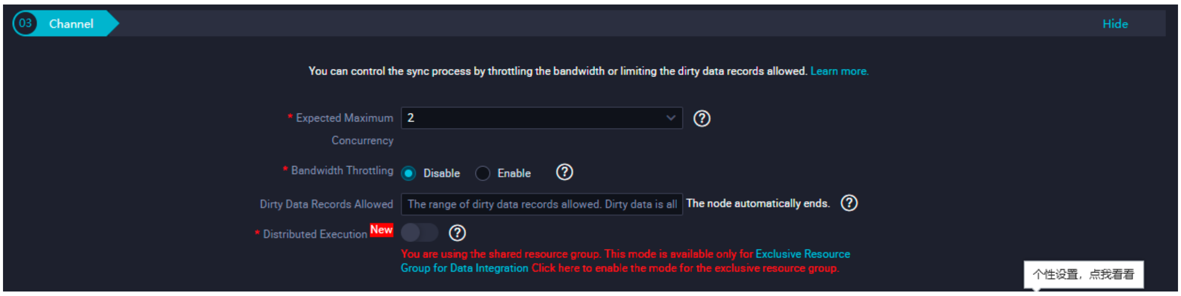
Field	Type
education	STRING
num	BIGINT

Target Table

Field	Type
education	STRING
num	BIGINT

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specified rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	Click **Add** to add a field. Take note of the following rules when you add a field: ◦ You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ◦ You can use scheduling parameters, such as ${bizdate}. ◦ If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to avoid heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.

Parameter	Description
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure MaxCompute Reader by using the code editor

In the following code, a synchronization node is configured to read data from MaxCompute. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

Notice Delete the comments from the following code before you run the code:

```
{
    "type":"job",
    "version":"2.0",
    "steps":[
        {
            "stepType":"odps",// The reader type.
            "parameter":{
                "partition":[],// The partitions from which you want to read data.
                "isCompress":false,// Specifies whether to enable compression.
                "datasource":"",// The name of the data source.
                "column":[// The names of the columns from which you want to read data.
                    "id"
                ],
                "emptyAsNull":true,
                "table":""// The name of the table from which you want to read data.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle": true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

If you want to specify the Tunnel endpoint of MaxCompute, you can use the code editor to configure the data source. To configure the data source, replace `"datasource":"",` in the preceding code with the parameters of the data source. The following code provides an example:

```
"accessId":"*******************",
"accessKey":"*******************",
"endpoint":"http://service.eu-central-1.maxcompute.aliyun-inc.com/api",
"odpsServer":"http://service.eu-central-1.maxcompute.aliyun-inc.com/api",
"tunnelServer":"http://dt.eu-central-1.maxcompute.aliyun.com",
"project":"*****",
```

5.2.26. Prometheus Reader

Prometheus is a time series database (TSDB) that is developed and maintained by SoundCloud. Prometheus is the open source implementation of BorgMon, the monitoring system of Google. Prometheus Reader can read data from Prometheus databases.

Notice

- Prometheus Reader supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.
- You can configure Prometheus Reader by using only the code editor.

How it works

Prometheus Reader connects to a Prometheus instance by sending an HTTP request and obtains raw data points by using the HTTP API endpoint `/api/v1/query_range`. A synchronization node is split into multiple tasks based on the metric and time range.

Limits

- The specified start time and end time are automatically converted to on-the-hour time. For example, if you set the time range to `[3:35, 4:55)` of April 18, 2019, the time range is converted to `[3:00, 4:00)`.
- Prometheus Reader supports only Prometheus 2.9.x.
- The time range that you specify is divided at a granularity of 10 seconds by default.

  The `/api/v1/query_range` endpoint allows you to query only a limited number of data points. If you specify an excessively large time range, the following error message is returned: `exceeded maximum resolution of 11,000 points per timeseries`. Therefore, Prometheus Reader obtains data at a granularity of 10 seconds by default. Even if the raw data points are stored by millisecond, you can query only a maximum of 10,000 data points by using the `/api/v1/query_range` endpoint.

Data types

The following table lists the mapping between the Data Integration data type and Prometheus data type.

Category	Data Integration data type	Prometheus data type
String	STRING	String to which a data point in Prometheus is serialized. The data point can be a timestamp, metric, tag, or value.

Parameters

Parameter	Description	Required	Default value
endpoint	The HTTP endpoint of Prometheus, in the format of http://IP address:Port number.	Yes	No default value
column	The metrics from which you want to read data points.	Yes	No default value

Parameter	Description	Required	Default value
beginDateTime	The start time of the time range of the data points that you want to read, in the format of yyyyMMddHHmmss. The beginDateTime parameter and the endDateTime parameter must be used in pairs.	Yes	No default value ⓘ **Note** The start time and end time of the time range are automatically converted to on-the-hour time. For example, if you set the time range to `[3:35, 4:55)` of April 18, 2019, the time range is converted to `[3:00, 4:00)` .
endDateTime	The end time of the time range of the data points that you want to read, in the format of yyyyMMddHHmmss. The endDateTime parameter and the beginDateTime parameter must be used in pairs.	Yes	No default value ⓘ **Note** The start time and end time of the time range are automatically converted to on-the-hour time. For example, if you set the time range to `[3:35, 4:55)` of April 18, 2019, the time range is converted to `[3:00, 4:00)` .

Configure Prometheus Reader by using the codeless UI

This method is not supported.

Configure Prometheus Reader by using the code editor

In the following code, a synchronization node is configured to read data from a Prometheus database. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "concurrent": 1,// The maximum number of parallel threads.
            "throttle": true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "steps": [
        {
            "category": "reader",
            "name": "Reader",
            "parameter": {
                "endpoint": "http://localhost:9090",
                "column": [
                    "up"
                ],
                "beginDateTime": "20190520150000",
                "endDateTime": "20190520160000"
            },
            "stepType": "prometheus"
        },
        {
            "category": "writer",
            "name": "Writer",
            "parameter": {},
            "stepType": ""
        }
    ],
    "type": "job",
    "version": "2.0"
}
```

Performance test report

Number of channels	Data integration speed (records/s)	Data integration bandwidth (Mbit/s)
1	45,000	5.36
2	55,384	6.60
3	60,000	7.15

5.2.27. PostgreSQL Reader

This topic describes the data types and parameters that are supported by PostgreSQL Reader and how to configure PostgreSQL Reader by using the codeless user interface (UI) and code editor.

Background information

PostgreSQL Reader reads data from PostgreSQL.

PostgreSQL Reader connects to a remote PostgreSQL database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the statement on the database and returns data. Then, PostgreSQL Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

- PostgreSQL Reader generates the SQL statement based on the settings of the table, column, and where parameters and sends the generated statement to the PostgreSQL database.
- If you specify the querySql parameter, PostgreSQL Reader directly sends the value of this parameter to the PostgreSQL database.

Precautions

If the name of a PostgreSQL table or a field in the table starts with a digit or contains letters or hyphens (-), you must use double quotation marks (") to escape the name. Otherwise, PostgreSQL Reader fails to read data from the PostgreSQL database. For PostgreSQL Reader, double quotation marks (") are keywords in JSON code. Therefore, you must use backslashes (\) to escape the double quotation marks ("). For example, if the name of a PostgreSQL table is `123Test`, the name becomes `\"123Test\"` after it is escaped.

Note

- Each double quotation mark (") must be escaped by using a backslash (\).
- You can use only the code editor to escape characters.

The following code provides an example on how to use the code editor to escape characters:

```
"parameter": {
    "datasource": "abc",
    "column": [
        "id",
        "\"123Test\"", // Add escape characters.
],
"where": "",
"splitPk": "id",
"table": "public.wpw_test"
},
```

Data types

PostgreSQL Reader supports most PostgreSQL data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by PostgreSQL Reader.

Category	PostgreSQL data type
Integer	BIGINT, BIGSERIAL, INTEGER, SMALLINT, and SERIAL
Floating point	DOUBLE, PRECISION, MONEY, NUMERIC, and REAL
String	VARCHAR, CHAR, TEXT, BIT, and INET
Date and time	DATE, TIME, and TIMESTAMP
Boolean	BOOLEAN
Binary	BYTEA

Note

- PostgreSQL Reader supports only the data types that are listed in the preceding table.
- You can convert the MONEY, INET, and BIT data types by using syntax such as `a_inet::varchar`.

Parameters

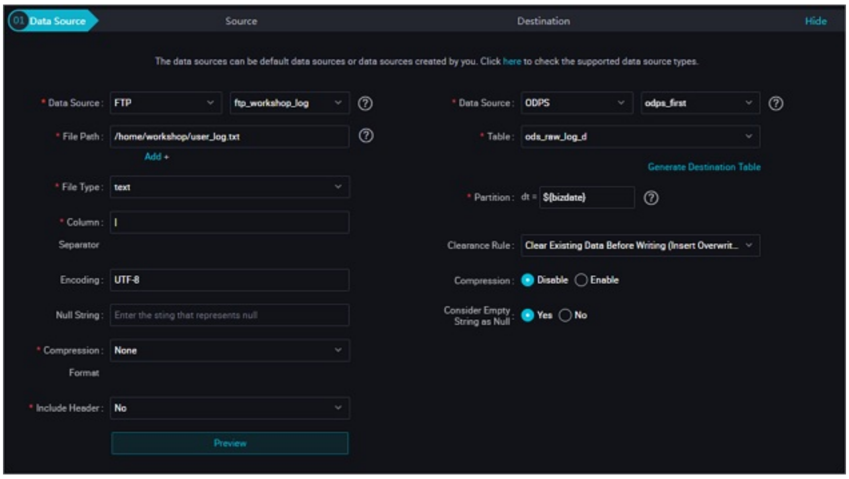
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table from which you want to read data.	Yes	No default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. The default value is [*], which indicates all the columns in the source table. • You can select specific columns to read. • The column order can be changed. This indicates that you can specify columns in an order different from the order specified by the schema of the source table. • Constants are supported. The column names must be arranged in compliance with the SQL syntax that is supported by PostgreSQL, such as `["id", "table","1", "'mingya.wmy'", "'null'", "to_char(a+1)", "2.3" , "true"]` . ◦ id: a column name. ◦ table: the name of a column that contains reserved keywords. ◦ 1: an integer constant. ◦ 'mingya.wmy': a string constant, which is enclosed in single quotation marks ('). ◦ 'null': a string. ◦ to_char(a+1): a function expression that is used to calculate the length of a string. ◦ 2.3: a floating-point constant. ◦ true: a Boolean value. • The column parameter must explicitly specify all the columns from which you want to read data. The parameter cannot be left empty.	Yes	No default value
splitPk	The field that is used for data sharding when PostgreSQL Reader reads data. If you specify this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This way, data can be synchronized more efficiently. • We recommend that you set the splitPk parameter to the name of the primary key column of the table. Data can be evenly distributed to different shards based on the primary key column, but not intensively distributed only to specific shards. • The splitPk parameter supports sharding for data only of integer data types. If you set this parameter to a column of an unsupported data type, such as a string, floating point, or date data type, PostgreSQL Reader ignores the setting of the splitPk parameter and uses a single thread to read data. • If the splitPk parameter is not provided or is left empty, PostgreSQL Reader uses a single thread to read data.	No	No default value
where	The WHERE clause. PostgreSQL Reader generates an SQL statement based on the settings of the table, column, and where parameters and uses the generated statement to read data. For example, you can set this parameter to `id>2 and sex=1` in an actual business scenario to synchronize the data that is generated on the current day. • You can use the WHERE clause to read incremental data. • If the where parameter is not provided or is left empty, PostgreSQL Reader reads all data.	No	No default value

Parameter	Description	Required	Default value
querySql (advanced parameter, which is available only in the code editor)	The SQL statement that is used for refined data filtering. If you specify this parameter, Data Integration filters data based on the value of this parameter. For example, if you want to join multiple tables for data synchronization, set this parameter to `select a,b from table_a join table_b on table_a.id = table_b.id` . If you specify this parameter, PostgreSQL Reader ignores the settings of the table, column, and where parameters.	No	No default value
fetchSize	The number of data records to read at a time. This parameter determines the number of interactions between Data Integration and the database and affects read efficiency. **Note** If you set this parameter to a value greater than 2048, an out of memory (OOM) error may occur during data synchronization.	No	*512*

Configure PostgreSQL Reader by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



The data sources can be default data sources or data sources created by you. Click here to check the supported data source types.

Source:

- * Data Source : FTP ftp_workshop_log
- * File Path : /home/workshop/user_log.txt
- * File Type : text
- * Column Separator : |
- Encoding : UTF-8
- Null String : Enter the sting that represents null
- * Compression Format : None
- * Include Header : No

Destination:

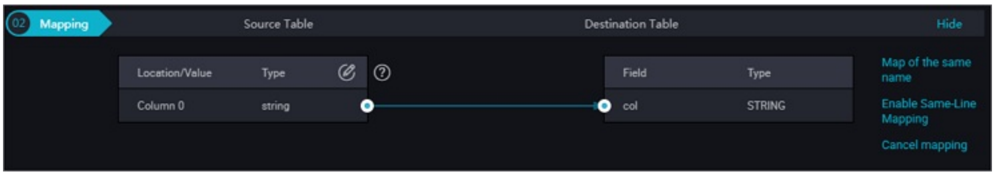
- * Data Source : ODPS odps_first
- * Table : ods_raw_log_d
- * Partition : dt = ${bizdate}
- Clearance Rule : Clear Existing Data Before Writing (Insert Overwrit...
- Compression : ● Disable ○ Enable
- Consider Empty String as Null : ● Yes ○ No

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table from which you want to read data. This parameter is equivalent to the table parameter that is described in the preceding section.
Filter	The condition used to filter the data that you want to read. Filtering based on the LIMIT keyword is not supported. The SQL syntax is determined by the selected data source.
Shard Key	The shard key. You can use a column in the source table as the shard key. We recommend that you use the primary key column or an indexed column. Only integer columns are supported. If you specify this parameter, data sharding is performed based on the value of this parameter, and parallel threads can be used to read data. This improves data synchronization efficiency. **Note** The Shard Key parameter is displayed only after you select the data source for the synchronization node.

2. Configure field mappings. This operation is equivalent to setting the **column** parameter that is described in the

preceding section.

Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



02 Mapping

Source Table

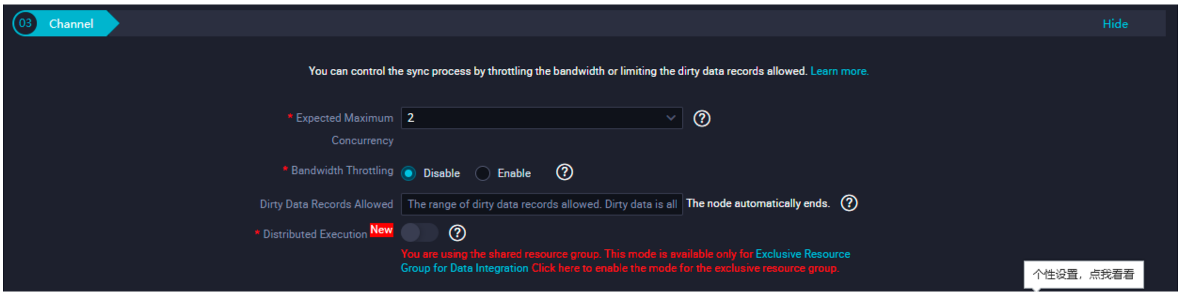
Location/Value	Type
Column 0	string

Destination Table

Field	Type
col	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.
Change Fields	You can manually edit fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	○ Click Add to add a field. Take note of the following rules when you add a field: You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ○ You can use scheduling parameters, such as ${bizdate}. ○ You can enter functions that are supported by relational databases, such as now() and count(1). ○ If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



03 Channel

You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.

Parameter	Description
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure PostgreSQL Reader by using the code editor

In the following code, a synchronization node is configured to read data from a PostgreSQL database. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"postgresql",// The reader type.
            "parameter":{
                "datasource":"",// The name of the data source.
                "column":[// The names of the columns from which you want to read data.
                    "col1",
                    "col2"
                ],
                "where":"",// The WHERE clause.
                "splitPk":"",// The shard key based on which the table is sharded. Data Integration runs parall
el threads to synchronize data.
                "table":""// The name of the source table.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":false, // Specifies whether to enable bandwidth throttling. The value false indicates th
at bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbp
s parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

Additional information

- Data synchronization between primary and secondary databases

  A secondary PostgreSQL database can be deployed for disaster recovery. The secondary database continuously synchronizes data from the primary database based on binary logs. Data latency between the primary and secondary databases cannot be prevented. This may result in data inconsistency.

- Data consistency control

  PostgreSQL is a relational database management system (RDBMS) that supports strong consistency for data queries. A database snapshot is created before a synchronization node starts. PostgreSQL Reader reads data from the database snapshot. Therefore, if new data is written to the database during data synchronization, PostgreSQL Reader cannot obtain the new data.

Data consistency cannot be ensured if you enable PostgreSQL Reader to use parallel threads to read data in a synchronization node.

PostgreSQL Reader shards the source table based on the value of the splitPk parameter and uses parallel threads to read data. These parallel threads belong to different transactions. They read data at different points in time. Therefore, the parallel threads observe different snapshots.

Data inconsistencies cannot be prevented if parallel threads are used for a synchronization node. The following workarounds can be used:

  - Enable PostgreSQL Reader to use a single thread to read data in a synchronization node. This indicates that you do not need to specify a shard key for PostgreSQL Reader. This way, data consistency is ensured, but data is synchronized at low efficiency.
  - Make sure that no data is written to the source table during data synchronization. This ensures that the data in the source table remains unchanged during data synchronization. For example, you can lock the source table or disable data synchronization between primary and secondary databases. This way, data is efficiently synchronized, but your ongoing services may be interrupted.

- Character encoding

  A PostgreSQL database supports only the EUC_CN and UTF-8 encoding formats for simplified Chinese characters. PostgreSQL Reader uses JDBC to read data. This enables PostgreSQL Reader to automatically convert the encoding formats of characters. Therefore, you do not need to specify the encoding format.

  If you specify the encoding format for a PostgreSQL database but data is written to the PostgreSQL database in a different encoding format, PostgreSQL Reader cannot identify this inconsistency and may export garbled characters.

- Incremental data synchronization

  PostgreSQL Reader uses JDBC to connect to a database and uses a SELECT statement with a `WHERE` clause to read incremental data.

  - For batch data, incremental add, update, and delete operations (including logically delete operations) are distinguished by timestamps. Specify the WHERE clause based on a specific timestamp. The time indicated by the timestamp must be later than the time indicated by the latest timestamp in the previous synchronization.
  - For streaming data, specify the WHERE clause based on the ID of a specific record. The ID must be greater than the maximum ID involved in the previous synchronization.

  If the data that is added or modified cannot be distinguished, PostgreSQL Reader can read only full data.

- Syntax validation

  PostgreSQL Reader allows you to configure custom SELECT statements by using the querySql parameter but does not verify the syntax of these statements.

5.2.28. OTSStream Reader

This topic describes the data types and parameters that are supported by OTSStream Reader and how to configure OTSStream Reader by using the codeless user interface (UI) and code editor.

Background information

OTSStream Reader reads incremental data from Tablestore. Incremental data can be considered as operation logs that include data and operation information.

Unlike the readers that are used to read full data, OTSStream Reader supports only the multi-version mode. If you use OTSStream Reader to read incremental data, you cannot specify the columns from which you want to read data. Before you use OTSStream Reader, make sure that the Stream feature is enabled for your source table. You can enable the Stream feature for a table when you create the table. You can also call the UpdateTable operation in the Tablestore SDK to enable this feature after a table is created.

The following sample code provides an example on how to enable the Stream feature:

```
SyncClient client = new SyncClient("", "", "", "");
Enable the Stream feature when you create a table.
CreateTableRequest createTableRequest = new CreateTableRequest(tableMeta);
createTableRequest.setStreamSpecification(new StreamSpecification(true, 24)); // The value 24 indicates that Ta
blestore retains incremental data for 24 hours.
client.createTable(createTableRequest);
If you do not enable the Stream feature when you create a table, you can call the UpdateTable operation to enab
le this feature after the table is created.
UpdateTableRequest updateTableRequest = new UpdateTableRequest("tableName");
updateTableRequest.setStreamSpecification(new StreamSpecification(true, 24));
client.updateTable(updateTableRequest);
```

You can enable the Stream feature and specify expiration time for incremental data by calling the UpdateTable operation in the Tablestore SDK. After you enable the Stream feature, the Tablestore server stores the operation logs of your Tablestore table. Each partition queues operation logs in sequence. Each operation log is recycled after the specified expiration time.

The Tablestore SDK provides several Stream-related API operations that are used to read operation logs. OTSStream Reader calls these API operations to read incremental data. By default, OTSStream Reader converts the incremental data into multiple 6-tuples and sends them to a writer. Each 6-tuple consists of pk, colName, version, colValue, opType, and sequenceInfo.

Read data by column

In the multi-version mode of Tablestore, table data is organized in a three-level structure: row, column, and version. One row can have multiple columns, and the column name is not fixed. Each column can have multiple versions, and each version has a specific timestamp, which is the version number.

You can call the API operations of Tablestore to read and write data. Tablestore records the recent write and modify operations that are performed on table data to record the incremental data. Therefore, incremental data can be considered as a set of operation records.

Tablestore supports the following types of operations:

- PutRow: writes a row. If the row exists, it is overwritten.
- UpdateRow: updates a row without modifying other data of the original row. You can use UpdateRow to add column values, overwrite column values if a version of a column exists, or delete a specific version or all the versions of a column.
- DeleteRow: deletes a row.

Tablestore generates incremental data records based on each type of operation. OTSStream Reader reads these records and converts them to a format supported by Data Integration.

Tablestore supports dynamic columns and the multi-version mode. Therefore, a row that is generated by OTSStream Reader is a version of a column rather than a row in Tablestore. After OTSStream Reader reads data from a row in Tablestore, OTSStream Reader converts the data to multiple rows. Each row includes the primary key value, column name, timestamp of the version for the column (version number), value of the version, and operation type. If the isExportSequenceInfo parameter is set to true, time series information is also included.

The following types of operations are defined for the data that is converted to the format supported by Data Integration:

- U (UPDATE): writes a version of a column.
- DO (DELETE_ONE_VERSION): deletes a version of a column.
- DA (DELETE_ALL_VERSION): deletes all the versions of a column based on the primary key value and the column name.
- DR (DELETE_ROW): deletes a row based on the primary key value.

The following table lists the data that is converted by OTSStream Reader from a table that has two primary key columns (pkName1 and pkName2).

pkName1	pkName2	columnName	timestamp	columnValue	opType
pk1_V1	pk2_V1	col_a	1441803688001	col_val1	U
pk1_V1	pk2_V1	col_a	1441803688002	col_val2	U
pk1_V1	pk2_V1	col_b	1441803688003	col_val3	U
pk1_V2	pk2_V2	col_a	1441803688000	-	DO

pkName1	pkName2	columnName	timestamp	columnValue	opType
pk1_V2	pk2_V2	col_b	-	-	DA
pk1_V3	pk2_V3	-	-	-	DR
pk1_V3	pk2_V3	col_a	1441803688005	col_val1	U

In the preceding example, three rows in the Tablestore table are read and converted to seven rows. The primary keys for the three rows are (pk1_V1, pk2_V1), (pk1_V2, pk2_V2), and (pk1_V3, pk2_V3).

- For the row whose primary key is (pk1_V1, pk2_V1), two versions of the col_a column and one version of the col_b column are written.
- For the row whose primary key is (pk1_V2, pk2_V2), one version of the col_a column and all the versions of the col_b column are deleted.
- For the row whose primary key is (pk1_V3, pk2_V3), one version of the col_a column is written, and the row is deleted.

Read data by row

You can also use OTSStream Reader to read data by row. In this mode, OTSStream Reader reads operation records as rows. You must set the mode parameter and specify the columns whose data you want to read.

```
"parameter": {
  # Set mode to single_version_and_update_only, set isExportSequenceInfo to false, and configure other paramete
rs, such as datasource and table, based on your business requirements.
  "mode": "single_version_and_update_only", # The read mode.
  "column":[  # The columns whose data you want to read from Tablestore. You can specify the columns based on y
our business requirements.
          {
             "name": "uid"  # The name of a column, which can be a primary key column or a property column.
          },
          {
             "name": "name"  # The name of a column, which can be a primary key column or a property column.
          },
  ],
  "isExportSequenceInfo": false, # Specifies whether to read time series information. If you set the mode param
eter to single_version_and_update_only, this parameter can only be set to false.
}
```

The data that is read by row is closer to the data in the original rows. This facilitates further data processing. If you read data by row, take note of the following points:

- The rows that are read are extracted from operation records. Each row corresponds to a write or update operation. If you update only some of the columns for a row, the operation record contains only the columns that are updated.
- The version number of each column, which is the timestamp of each column, cannot be read or deleted.

Data types

OTSStream Reader supports all Tablestore data types. The following table lists the data types that are supported by OTSStream Reader.

Category	Tablestore data type
Integer	INTEGER
Floating point	DOUBLE
String	STRING
Boolean	BOOLEAN
Binary	BINARY

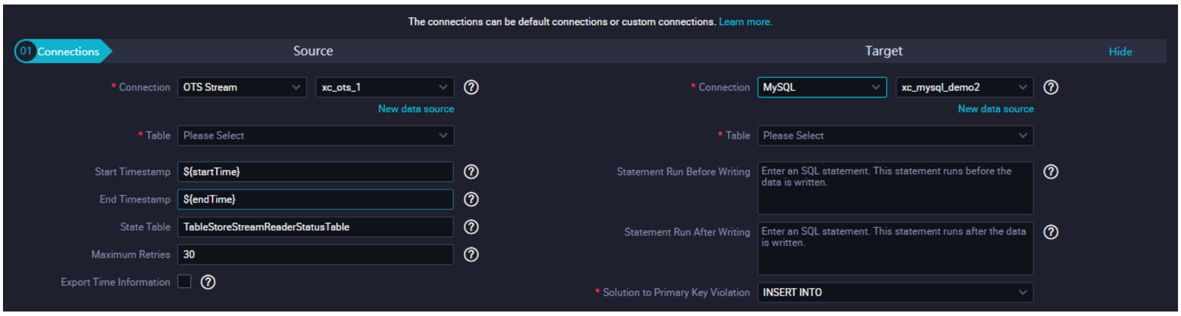
Parameters

Parameter	Description	Required	Default value
dataSource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
dataTable	The name of the table from which you want to read incremental data. You can enable the Stream feature for a table when you create the table. You can also call the UpdateTable operation to enable this feature after the table is created.	Yes	No default value
statusTable	The name of the table that OTSStream Reader uses to store status records. These records help find the data that is not required and improve read efficiency. If the specified table does not exist, OTSStream Reader automatically creates such a table. After an offline read task is completed, you do not need to delete the table. The status records in the table can be used for the next read task. • You do not need to create a status table. You need only to provide a table name. OTSStream Reader attempts to create a status table on your instance. If no such table exists, OTSStream Reader automatically creates one. If the table already exists, OTSStream Reader determines whether the metadata of the table meets the expectation. If the metadata does not meet the expectation, OTSStream Reader reports an error. • After a read task is completed, you do not need to delete the table. The status records in the table can be used for the next read task. • Time-to-live (TTL) is enabled for the table, and data automatically expires based on the TTL. This way, the table stores a small volume of data. • You can use the same status table to store the status records of multiple tables that are specified by the dataTable parameter and managed by the same instance. The status records are independent of each other. You can configure a name similar to TableStoreStreamReaderStatusTable. Make sure that the name is different from the name of a business-related table.	Yes	No default value
date	The date on which the data that you want to read is generated. Specify this parameter in the yyyyMMdd format, such as 20151111.	No	No default value
isExportSequenceInfo	Specifies whether to read time series information. The time series information includes the time when data is written. The default value is *false*, which indicates that time series information is not read.	No	*false*
maxRetries	The maximum number of retries for each request to read incremental data from Tablestore. Default value: 30. Retries are performed at a specific interval. The total duration of 30 retries is approximately 5 minutes. You can keep the default settings.	No	*30*
startTimeString	The start time of the incremental data, in seconds. The start time is the left boundary of the left-closed, right-open time range of the incremental data. Specify this parameter in the `yyyymmddhh24miss` format.	No	No default value
endTimeString	The end time of the incremental data, in seconds. The end time is the right boundary of the left-closed, right-open time range of the incremental data. Specify this parameter in the `yyyymmddhh24miss` format.	No	No default value
mode	The read mode. If this parameter is set to single_version_and_update_only, data is read by row.	No	No default value

Configure OTSStream Reader by using the codeless UI

1. Configure data sources.

Configure **Source** and **Target** for the synchronization node.



The connections can be default connections or custom connections. Learn more.

01 Connections

Source

* Connection: OTS Stream, xc_ots_1

New data source

* Table: Please Select

Start Timestamp: ${startTime}

End Timestamp: ${endTime}

State Table: TableStoreStreamReaderStatusTable

Maximum Retries: 30

Export Time Information: □

Target

* Connection: MySQL, xc_mysql_demo2

New data source

* Table: Please Select

Statement Run Before Writing: Enter an SQL statement. This statement runs before the data is written.

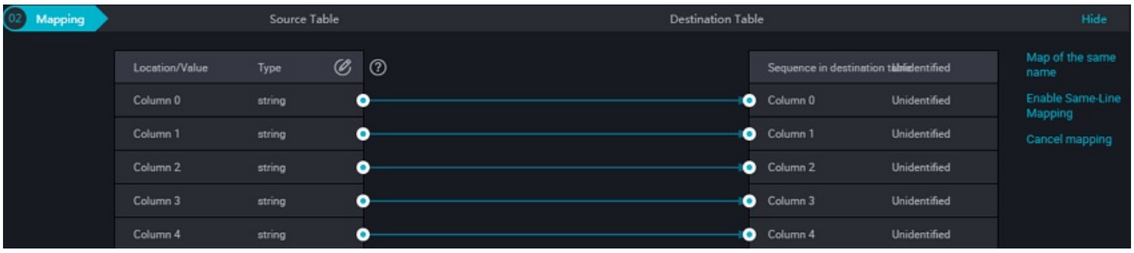
Statement Run After Writing: Enter an SQL statement. This statement runs after the data is written.

* Solution to Primary Key Violation: INSERT INTO

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the dataSource parameter that is described in the preceding section.
Table	The name of the table from which you want to read data. This parameter is equivalent to the dataTable parameter that is described in the preceding section.
Start Timestamp	The start time of the incremental data, in seconds. The start time is the left boundary of the left-closed, right-open time range of the incremental data. Specify this parameter in the `yyyymmddhh24miss` format. **Note** The start time must be a point in time that is within seven days from the time when the synchronization node is configured.
End Timestamp	The end time of the incremental data, in seconds. The end time is the right boundary of the left-closed, right-open time range of the incremental data. Specify this parameter in the `yyyymmddhh24miss` format.
State Table	The name of the table that is used to store status records.
Maximum Retries	The maximum number of retries. This parameter is equivalent to the maxRetries parameter that is described in the preceding section. Default value: 30.
Export Time Information	Specifies whether to read time series information. This parameter is equivalent to the isExportSequenceInfo parameter that is described in the preceding section. Default value: *false*.

2. Configure field mappings.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



02 Mapping

Source Table

Location/Value	Type
Column 0	string
Column 1	string
Column 2	string
Column 3	string
Column 4	string

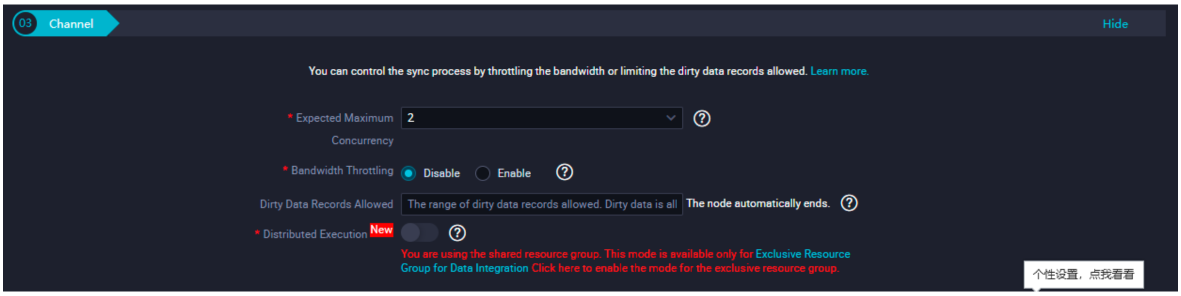
Destination Table

Sequence in destination table	Unidentified
Column 0	Unidentified
Column 1	Unidentified
Column 2	Unidentified
Column 3	Unidentified
Column 4	Unidentified

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.

Operation	Description
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure OTSStream Reader by using the code editor

In the following code, a synchronization node is configured to read data from Tablestore. For more information about the parameters, see the preceding parameter description. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"otsstream",// The reader type.
            "parameter":{
                "statusTable":"TableStoreStreamReaderStatusTable",// The name of the table that OTSStream Reader uses to store status records.
                "maxRetries":30,// The maximum number of retries for each request to read incremental data from Tablestore. Default value: 30.
                "isExportSequenceInfo":false,// Specifies whether to read time series information.
                "datasource":"$srcDatasource",// The name of the data source.
                "startTimeString":"${startTime}${hh}",// The start time of the incremental data, which is the same as the time when a synchronization node starts to run. Specify this parameter in the yyyymmddhh24miss format.
                "table":"",// The name of the table from which you want to read data.
                "endTimeString":"${endTime}${hh}"// The end time of the incremental data, which is the same as the time when the running of a synchronization node is complete. Specify this parameter in the yyyymmddhh24miss format.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1 // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.2.29. MetaQ Reader

This topic describes the data types and parameters that are supported by MetaQ Reader and how to configure MetaQ Reader by using the code editor.

Notice MetaQ Reader supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration, Use the default resource group, and Create a custom resource group for Data Integration.

Background information

Message Queue is a professional message-oriented middleware that is developed by Alibaba Group. Message Queue provides a complete set of cloud messaging services based on the technologies that are used for building a highly available and distributed cluster. The services include message subscription and publishing, message tracing, scheduled and delayed messages, resource statistics, and monitoring and alerting. Message Queue provides asynchronous decoupling for distributed application systems and is suitable for Internet applications with large amounts of messages and high throughput. It is one of the core services that are used by Alibaba Group to support the promotional events of Double 11.

MetaQ Reader reads real-time data from Message Queue by using Message Queue SDK for Java. Then, MetaQ Reader converts the data types of the data to those supported by Data Integration and sends the converted data to a writer.

How it works

MetaQ Reader subscribes to real-time data from Message Queue by using Message Queue SDK for Java of one of the following versions:

```
<dependency>
            <groupId>com.taobao.metaq.final</groupId>
            <artifactId>metaq-client</artifactId>
            <version>4.0.1</version>
        </dependency>
        <dependency>
            <groupId>com.aliyun.openservices</groupId>
            <artifactId>ons-sdk</artifactId>
            <version>1.3.1</version>
        </dependency>
```

Data types

The following table lists the mapping between Data Integration data type and Message Queue data type.

Data Integration data type	Message Queue data type
STRING	STRING

Parameters

Parameter	Description	Required
accessId	The AccessKey ID that you use to access Message Queue.	Yes
accessKey	The AccessKey secret that you use to access Message Queue.	Yes
consumerId	The consumer ID. A consumer is also known as a message subscriber which receives and consumes messages. The consumer ID is the identifier of a type of consumer. In most cases, the consumers that have the same consumer ID receive and consume the same type of message and use the same consumption logic.	Yes
topicName	The topic of the messages that you want to consume. A topic is used to classify messages. It is the primary classifier.	Yes
subExpression	The subtopic of the messages.	Yes

Parameter	Description	Required
onsChannel	The channel that is used for authentication when MetaQ Reader connects to Message Queue.	Yes
unitName	The destination unit that receives messages. Valid values: • sh: center • unsz: unit in the China (Shenzhen) region • us: United States of America • en-us: Europe • rg-ru: Russia • zbyk: Youku in the China (Zhangjiakou) region • unzbyun: Alibaba Cloud in the China (Zhangjiakou) region • unshyun: Alibaba Cloud in the China (Shanghai) region • lazada-sg: Lazada in Singapore • lazada-my: Lazada in Malaysia • lazada-vn: Lazada in Vietnam • lazada-ph: Lazada in the Philippines • lazada-th: Lazada in Thailand • lazada-id: Lazada in Indonesia	No
instanceName	The name of the consumer instance.	No
domainName	The endpoint that you use to connect to Message Queue.	Yes
contentType	The type of the messages. Valid values: *singlestringcolumn*, *text*, and *json*.	Yes
beginOffset	The offset from which MetaQ Reader starts to read data. Valid values: begin and lastRead.	Yes
nullCurrentOffset	The offset from which MetaQ Reader starts to read data if the last offset is null. Valid values: begin and current.	Yes
fieldDelimiter	The column delimiter that is used to separate message strings, such as commas (,). Control characters are supported. Example: *\u0001*.	Yes
column	The names of the fields from which you want to read data in the messages.	Yes
beginDateTime	The start time of data consumption. This parameter specifies the left boundary of a left-closed, right-open interval. The value of the beginDateTime parameter is a time string in the yyyyMMddHHmmss format. This parameter can be used together with the scheduling time parameters in DataWorks.	No

Parameter	Description	Required
endDateTime	The end time of data consumption. This parameter specifies the right boundary of a left-closed, right-open interval. The value of the endDateTime parameter is a time string in the yyyyMMddHHmmss format. This parameter can be used together with the scheduling time parameters in DataWorks.	**Note** The beginDateTime and endDateTime parameters must be used in pairs.

Configure MetaQ Reader by using the code editor

In the following code, a synchronization node is configured to read data from Message Queue. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "job": {
        "content": [
            {
                "reader": {
                    "name": "metaqreader",
                    "parameter": {
                        "accessId": "<yourAccessKeyId>",
                        "accessKey": "<yourAccessKeySecret>",
                        "consumerId": "Test01",
                        "topicName": "test",
                        "subExpression": "*",
                        "onsChannel": "ALIYUN",
                        "domainName": "***.aliyun.com",
                        "contentType": "singlestringcolumn",
                        "beginOffset": "lastRead",
                        "nullCurrentOffset": "begin",
                        "fieldDelimiter": ",",
                        "column": [
                            "col0"
                        ],
                        "fieldDelimiter": ","
                    }
                },
                "writer": {
                    "name": "streamwriter",
                    "parameter": {
                        "print": false
                    }
                }
            }
        ]
    }
}
```

5.2.30. Vertica Reader

Vertica is a column-oriented database that uses the massively parallel processing (MPP) architecture. Vertica Reader reads data from Vertica. This topic describes how Vertica Reader works, the parameters that are supported by Vertica Reader, and how to configure Vertica Reader by using the codeless user interface (UI) and code editor.

Notice Vertica Reader supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.

How it works

Vertica Reader connects to a remote Vertica database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the statement on the database and returns data. Then, Vertica Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

- Vertica Reader generates the SQL statement based on the settings of the table, column, and where parameters and sends the generated statement to the Vertica database.
- If you specify the querySql parameter, Vertica Reader directly sends the value of this parameter to the Vertica database.

Vertica Reader connects to a Vertica database by using the Vertica JDBC driver. You must make sure that the driver version is compatible with your Vertica database. Vertica Reader uses the Vertica JDBC driver of the following version:

```
<dependency>
    <groupId>com.vertica</groupId>
    <artifactId>vertica-jdbc</artifactId>
    <version>7.1.2</version>
</dependency>
```

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
jdbcUrl	The JDBC URL that is used to connect to the Vertica database. You can specify multiple JDBC URLs for a database. Specify JDBC URLs in a JSON array. If you specify multiple JDBC URLs, Vertica Reader verifies the connectivity of the URLs in sequence to find a valid URL. If no URL is valid, Vertica Reader returns an error. **Note** The jdbcUrl parameter must be included in the connection parameter. The value of the jdbcUrl parameter must comply with the standard format that is supported by Vertica. You can also specify the information of the attachment facility. Example: `jdbc:vertica://1**.0.0.1:3306/database`.	No	No default value
username	The username that is used to connect to the database.	No	No default value
password	The password that is used to connect to the database.	No	No default value
table	The name of the table from which you want to read data. Vertica Reader can read data from multiple tables. Specify the table names in a JSON array. If you specify multiple tables, make sure that the tables have the same schema. Vertica Reader does not check whether the tables have the same schema. **Note** The table parameter must be included in the connection parameter.	Yes	No default value

Parameter	Description	Required	Default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. The default value is [*], which indicates all the columns in the source table. - You can select specific columns to read. - The column order can be changed. This indicates that you can specify columns in an order different from the order specified by the schema of the source table. - Constants are supported. - The column parameter must explicitly specify all the columns from which you want to read data. The parameter cannot be left empty.	Yes	No default value
splitPk	The field that is used for data sharding when Vertica Reader reads data. If you specify this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This improves data synchronization efficiency. - We recommend that you set the splitPk parameter to the name of the primary key column of the table. Data can be evenly distributed to different shards based on the primary key column, instead of being intensively distributed only to specific shards. - The splitPk parameter supports sharding only for data of integer data types. If you set this parameter to a field of an unsupported data type, such as a string, floating point, or date data type, Vertica Reader returns an error. - If the splitPk parameter is left empty, Vertica Reader uses a single thread to read data from the source table.	No	No default value
where	The WHERE clause. Vertica Reader generates an SQL statement based on the settings of the table, column, and where parameters and uses the generated statement to read data. For example, when you perform a test, you can specify the where parameter to filter data. In actual business scenarios, you can set the where parameter to `gmt_create > $bizdate` to read the data that is generated on the current day. - You can use the WHERE clause to read incremental data. - If the where parameter is not provided or is left empty, Vertica Reader reads all data.	No	No default value
querySql	The SQL statement that is used for refined data filtering. If you specify this parameter, Data Integration filters data based only on the value of this parameter. If you specify the querySql parameter, Vertica Reader ignores the settings of the table, column, and where parameters.	No	No default value
fetchSize	The number of data records to read at a time. This parameter determines the number of interactions between Data Integration and Vertica and affects read efficiency. **Note** If you set this parameter to a value greater than 2048, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

Configure Vertica Reader by using the codeless UI

This method is not supported.

Configure Vertica Reader by using the code editor

In the following code, a synchronization node is configured to read data from Vertica. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type": "job",
    "steps": [
        {
            "stepType": "vertica", // The reader type.
            "parameter": {
                "datasource": "", // The name of the data source.
                "username": "",
                "password": "",
                "where": "",
                "column": [ // The names of the columns from which you want to read data.
                    "id",
                    "name"
                ],
                "splitPk": "id",
                "connection": [
                    {
                        "table": [ // The name of the table from which you want to read data.
                            "table"
                        ],
                        "jdbcUrl": [
                            "jdbc:vertica://host:port/database"
                        ]
                    }
                ]
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "stream",
            "parameter": {
                "print": false,
                "fieldDelimiter": ","
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0" // The maximum number of dirty data records allowed.
        },
        "speed": {
            "throttle": true, // Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent": 1, // The maximum number of parallel threads.
                        "mbps":"12"// The maximum transmission rate.
        }
    }
}
```

5.2.31. Gbase8a Reader

This topic describes the parameters that are supported by GBase8a Reader and how to configure GBase8a Reader by using the codeless user interface (UI) and code editor.

Background information

GBase 8a is a new type of column-oriented analytical database. GBase8a Reader can read data from GBase 8a databases.

Notice GBase8a Reader supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.

GBase8a Reader connects to a remote GBase 8a database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the statement on the database and returns data. Then, GBase8a Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

- GBase8a Reader generates the SQL statement based on the settings of the table, column, and where parameters and sends the generated statement to the GBase 8a database.
- If you specify the querySql parameter, GBase8a Reader sends the value of this parameter to the GBase 8a database.

GBase8a Reader uses the MySQL database driver to access a GBase 8a database. You must make sure that your GBase 8a database is compatible with the driver version. The driver used by GBase8a Reader is of the following version:

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.22</version>
</dependency>
```

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. If the edition of the DataWorks service that you activated supports GBase 8a data sources, you can add a GBase 8a data source and specify the data source in this parameter. You can connect to the added GBase 8a data source based on the setting of the jdbcUrl or username parameter.	No	No default value
jdbcUrl	The JDBC URL that is used to connect to the source database. You can specify multiple JDBC URLs in a JSON array for a database. If you specify multiple JDBC URLs, GBase8a Reader verifies the connectivity of the URLs in sequence to find a valid URL. If no URL is valid, GBase8a Reader returns an error. **Note** The jdbcUrl parameter must be included in the connection parameter. The value of the jdbcUrl parameter must comply with the standard format that is supported by GBase 8a. You can also specify the information of the attachment facility. An example JDBC URL is `jdbc:mysql://127.0.0.1:3306/database` . You must specify either the jdbcUrl or username parameter.	No	No default value
username	The username that is used to connect to the source database.	No	No default value

Parameter	Description	Required	Default value
password	The password that is used to connect to the source database.	No	No default value
table	The name of the table from which you want to read data. GBase8a Reader can read data from multiple tables. Specify the table names in a JSON array. If you specify multiple tables, make sure that the tables have the same schema. GBase8a Reader does not check whether the tables have the same schema. Note The table parameter must be included in the connection parameter.	Yes	No default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. The default value is [*], which indicates all the columns in the source table. - You can select specific columns to read. - The column order can be changed. This indicates that you can specify columns in an order different from the order specified by the schema of the source table. - Constants are supported, such as `'123'` . - Functions are supported, such as `date('now')` . - The column parameter must explicitly specify all the columns from which you want to read data. The parameter cannot be left empty.	Yes	No default value
splitPk	The field that is used for data sharding when GBase8a Reader reads data. If you specify this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This way, data can be synchronized more efficiently. - We recommend that you set the splitPk parameter to the name of the primary key column of the table. Data can be evenly distributed to different shards based on the primary key column, instead of being intensively distributed only to specific shards. - The splitPk parameter supports sharding for data only of integer data types. If you set the splitPk parameter to a field of an unsupported data type, such as a string, floating point, or date data type, the setting of this parameter is ignored, and a single thread is used to read data. - If you leave the splitPk parameter empty, a single thread is used to read data.	No	No default value
where	The WHERE clause. GBase8a Reader generates an SQL statement based on the settings of the column, table, and where parameters and uses the generated statement to read data. For example, when you perform a test, you can set the where parameter to limit 10. To read the data that is generated on the current day, you can set the where parameter to gmt_create > $bizdate. - You can use the WHERE clause to read incremental data. - If the where parameter is not provided or is left empty, GBase8a Reader reads all data.	No	No default value

Parameter	Description	Required	Default value
querySql	The SQL statement that is used for refined data filtering. If you specify this parameter, data is filtered based only on the value of this parameter. If you specify the querySql parameter, GBase8a Reader ignores the settings of the table, column, where, and splitPk parameters.	No	No default value
fetchSize	The number of data records to read at a time. This parameter determines the number of interactions between GBase8a Reader and the source database and affects read efficiency. **Note** If you set this parameter to a value greater than 2048, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

Configure GBase8a Reader by using the codeless UI

This method is not supported.

Configure GBase8a Reader by using the code editor

In the following code, a synchronization node is configured to read data from a GBase 8a database. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type": "job",
    "steps": [
        {
            "stepType": "gbase8a", // The reader type.
            "parameter": {
                "datasource": "", // The name of the data source.
                "username": "",
                "password": "",
                "where": "",
                "column": [ // The names of the columns from which you want to read data.
                    "id",
                    "name"
                ],
                "splitPk": "id",
                "connection": [
                    {
                        "table": [ // The name of the table from which you want to read data.
                            "table"
                        ],
                        "jdbcUrl": [
                            "jdbc:mysql://host:port/database"
                        ]
                    }
                ]
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "stream",
            "parameter": {
                "print": false,
                "fieldDelimiter": ","
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0" // The maximum number of dirty data records allowed.
        },
        "speed": {
            "throttle": true, // Specifies whether to enable bandwidth throttling. The value false indicates th
at bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbp
s parameter takes effect only when the throttle parameter is set to true.
            "concurrent": 1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    }
}
```

5.2.32. DataHub Reader

Alibaba Cloud DataHub is a streaming data processing platform. You can publish and subscribe to streaming data in DataHub and distribute the data to other platforms. This allows you to analyze streaming data and build applications based on the streaming data.

DataHub Reader reads data from DataHub by using the following version of SDK for Java:

```
<dependency>
    <groupId>com.aliyun.DataHub</groupId>
    <artifactId>aliyun-sdk-DataHub</artifactId>
    <version>2.9.1</version>
</dependency>
```

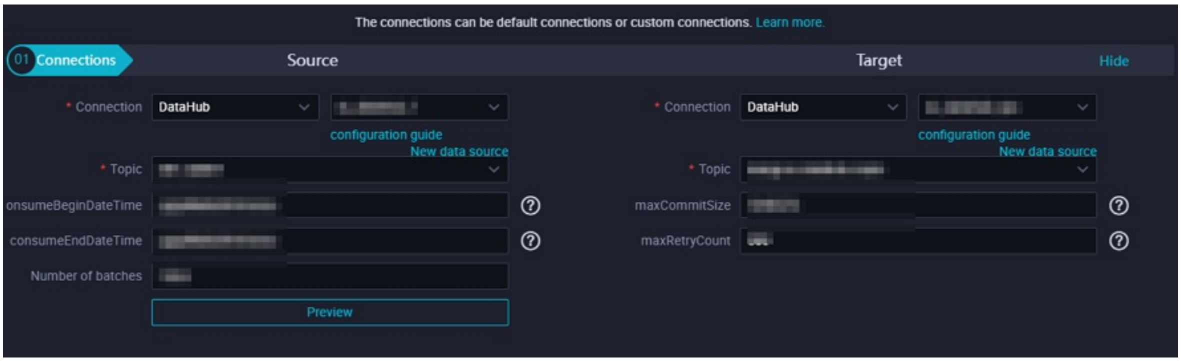
Parameters

Parameter	Description	Required
endpoint	The endpoint of DataHub.	Yes
accessId	The AccessKey ID that is used to connect to DataHub.	Yes
accessKey	The AccessKey secret that is used to connect to DataHub.	Yes
project	The name of the DataHub project from which you want to read data. DataHub projects are the resource management units in DataHub for resource isolation and control.	Yes
topic	The name of the DataHub topic from which you want to read data.	Yes
batchSize	The number of data records to read at a time. Default value: *1024*.	No
beginDateTime	The start time of data consumption. This parameter specifies the left boundary of a left-closed, right-open interval. Specify the start time in the format of yyyyMMddHHmmss. The parameter can be used together with the scheduling time parameters in DataWorks. **Note** The beginDateTime and endDateTime parameters must be used in pairs.	Yes
endDateTime	The end time of data consumption. This parameter specifies the right boundary of a left-closed, right-open interval. Specify the end time in the format of yyyyMMddHHmmss. The parameter can be used together with the scheduling time parameters in DataWorks. **Note** The beginDateTime and endDateTime parameters must be used in pairs.	Yes

Configure DataHub Reader by using the codeless UI

1. Configure data sources.

   Configure **Source** and Target for the synchronization node.



The connections can be default connections or custom connections. Learn more.

01 Connections — Source — Target — Hide

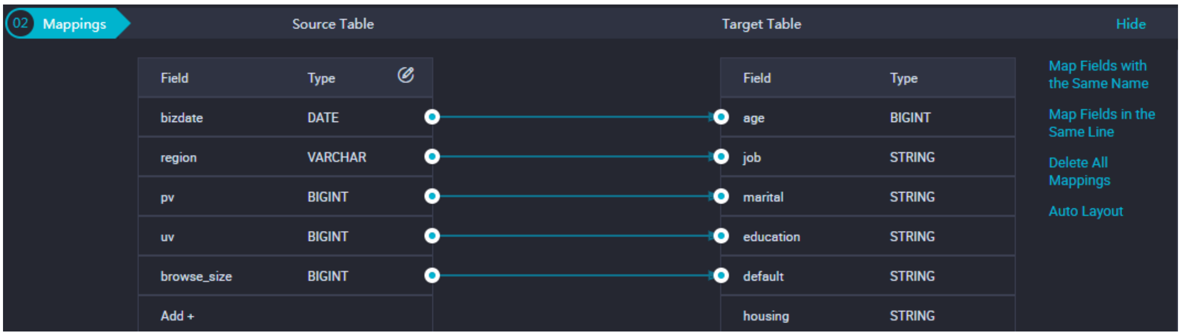
Source:
- * Connection: DataHub, [redacted]
- configuration guide
- New data source
- * Topic: [redacted]
- onsumeBeginDateTime: [redacted]
- consumeEndDateTime: [redacted]
- Number of batches: [redacted]
- Preview

Target:
- * Connection: DataHub, [redacted]
- configuration guide
- New data source
- * Topic: [redacted]
- maxCommitSize: [redacted]
- maxRetryCount: [redacted]

Parameter	Description
Connection	The name of the data source from which you want to read data.
Topic	This parameter is equivalent to the topic parameter that is described in the preceding section.
consumeBeginDateTime	This parameter is equivalent to the beginDateTime parameter that is described in the preceding section.
consumeEndDateTime	This parameter is equivalent to the endDateTime parameter that is described in the preceding section.
Number of batches	This parameter is equivalent to the batchSize parameter that is described in the preceding section.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



02 Mappings — Source Table — Target Table — Hide

Source Table:

Field	Type
bizdate	DATE
region	VARCHAR
pv	BIGINT
uv	BIGINT
browse_size	BIGINT
Add +	

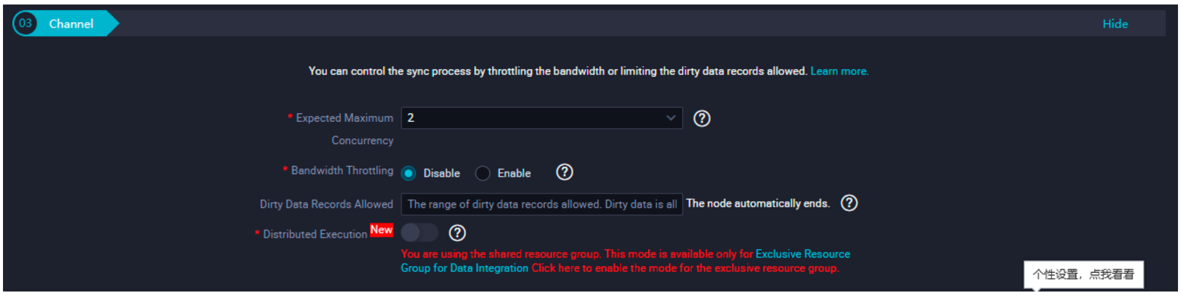
Target Table:

Field	Type
age	BIGINT
job	STRING
marital	STRING
education	STRING
default	STRING
housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

Operation	Description
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	Click **Add** to add a field. Take note of the following rules when you add a field: ◦ You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ◦ You can use scheduling parameters, such as ${bizdate}. ◦ You can enter functions that are supported by relational databases, such as now() and count(1). ◦ If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure DataHub Reader by using the code editor

In the following code, a synchronization node is configured to read data from DataHub. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "job": {
        "content": [
            {
                "reader": {
                    "name": "DataHubreader",
                    "parameter": {
                        "endpoint": "xxx" // The endpoint of DataHub.
                        "accessId": "xxx", // The AccessKey ID that is used to connect to DataHub.
                        "accessKey": "xxx", // The AccessKey secret that is used to connect to DataHub.
                        "project": "xxx", // The name of the DataHub project from which you want to read data.
                        "topic": "xxx" // The name of the DataHub topic from which you want to read data.
                        "batchSize": 1000, // The number of data records to read at a time.
                        "beginDateTime": "20180910111214", // The start time of data consumption.
                        "endDateTime": "20180910111614", // The end time of data consumption.
                        "column": [
                            "col0",
                            "col1",
                            "col2",
                            "col3",
                            "col4"
                        ]
                    }
                },
                "writer": {
                    "name": "streamwriter",
                    "parameter": {
                        "print": false
                    }
                }
            }
        ]
    }
}
```

5.2.33. ApsaraDB for OceanBase Reader

ApsaraDB for OceanBase is a financial-grade distributed relational database that is developed by Alibaba Cloud and Ant Financial. This topic describes the parameters that are supported by ApsaraDB for OceanBase Reader and how to configure ApsaraDB for OceanBase Reader by using the codeless user interface (UI) and code editor.

Notice ApsaraDB for OceanBase Reader supports only exclusive resource groups for Data Integration, but not shared resource groups or custom resource groups. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.

Background information

ApsaraDB for OceanBase implements automated and non-disruptive disaster recovery across cities based on the Five Data Centers Across Three Regions solution. It provides high availability for financial services based on conventional hardware. ApsaraDB for OceanBase is a database service developed by Alibaba Cloud. It has undergone strict verification in terms of functionality, stability, scalability, and performance.

ApsaraDB for OceanBase Reader reads data from tables stored in ApsaraDB for OceanBase databases.

ApsaraDB for OceanBase Reader connects to a remote ApsaraDB for OceanBase database by using a Java client, generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the statement on the database and returns data. Then, ApsaraDB for OceanBase Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

- ApsaraDB for OceanBase Reader generates an SQL statement based on the table, column, and where parameters that you have configured and sends the generated statement to the ApsaraDB for OceanBase database.
- If you set the querySql parameter, ApsaraDB for OceanBase Reader directly sends the value of this parameter to the ApsaraDB for OceanBase database.

> **Note** ApsaraDB for OceanBase supports the Oracle and MySQL tenant modes. Make sure that the WHERE clause and the columns that you specify in the columns parameter comply with the SQL syntax constraints that Oracle or MySQL supports. Otherwise, the SQL statement may fail to be executed.

ApsaraDB for OceanBase Reader accesses an ApsaraDB for OceanBase database by using the OceanBase driver. Confirm the compatibility between the driver version and your ApsaraDB for OceanBase database. ApsaraDB for OceanBase Reader uses the following version of the OceanBase database driver:

```
<dependency>
    <groupId>com.alipay.OceanBase</groupId>
    <artifactId>OceanBase-connector-java</artifactId>
    <version>3.1.0</version>
</dependency>
```

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the ApsaraDB for OceanBase database that you added in DataWorks. You can connect to the ApsaraDB for OceanBase database based on the settings of the jdbcUrl or username parameter.	Yes	No default value
jdbcUrl	The JDBC URL of the source database. You can specify multiple JDBC URLs in a JSON array for a database. If you specify multiple JDBC URLs, ApsaraDB for OceanBase Reader verifies the connectivity of the URLs in sequence to find a valid URL. If no URL is valid, ApsaraDB for OceanBase Reader returns an error. **Note** The jdbcUrl parameter must be included in the connection parameter. The value of the jdbcUrl parameter must be in compliance with the standard format that ApsaraDB for OceanBase supports. You can also specify the information of the attachment facility. An example JDBC URL is `jdbc:OceanBase://127.0.0.1:3306/database`. You must specify either jdbcUrl or username.	Yes	No default value
username	The username that is used to connect to the ApsaraDB for OceanBase database.	Yes	No default value
password	The password that is used to connect to the ApsaraDB for OceanBase database.	Yes	No default value
table	The name of the table from which you want to read data. ApsaraDB for OceanBase Reader can read data from multiple tables. The tables are specified in a JSON array. If you specify multiple tables, make sure that the tables have the same schema. ApsaraDB for OceanBase Reader does not check whether the tables have the same schema. **Note** The table parameter must be included in the connection parameter.	Yes	No default value

Parameter	Description	Required	Default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. The default value is [*], which indicates all the columns. - You can select specific columns to read. - The column order can be changed. This indicates that you can specify columns in an order different from the order specified by the schema of the source table. - Constants are supported. Example: `'123'` . - Functions are supported. Example: `date('now')` . - The column parameter must explicitly specify all the columns from which you want to read data. The parameter cannot be left empty.	Yes	No default value
splitPk	The field that is used for data sharding when ApsaraDB for OceanBase Reader reads data. If you specify this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This way, data can be synchronized more efficiently. - We recommend that you set the splitPk parameter to the name of the primary key column of the table. Data can be evenly distributed to different shards based on the primary key column, instead of being intensively distributed only to specific shards. - The splitPk parameter supports sharding for data only of integer data types. If you set this parameter to a field of an unsupported data type, such as a string, floating point, or date data type, ApsaraDB for OceanBase Reader returns an error. - If you leave the splitPk parameter empty, ApsaraDB for OceanBase Reader uses a single thread to read data.	Yes	Left empty
where	The WHERE clause. ApsaraDB for OceanBase Reader generates an SQL statement based on the column, table, and where parameters that you have configured and uses the generated statement to read data. For example, when you perform a test, you can set the where parameter to limit 10. You can set this parameter to gmt_create > $bizdate to read data on the current day. - You can use the WHERE clause to read incremental data. - If the where parameter is not provided or is left empty, ApsaraDB for OceanBase Reader reads all data.	Yes	No default value
querySql	The SQL statement that is used for refined data filtering. If you specify this parameter, Data Integration filters data based on the value of this parameter. If you specify this parameter, ApsaraDB for OceanBase Reader ignores the settings of the table, column, where, and splitPk parameters.	Yes	No default value
fetchSize	The number of data records to read at a time. This parameter determines the number of interactions between Data Integration and the database and affects read efficiency. **Note** If you set this parameter to a value greater than 2048, an out of memory (OOM) error may occur during data synchronization.	Yes	*1,024*

Configure ApsaraDB for OceanBase Reader by using the codeless UI

This method is not supported.

Configure ApsaraDB for OceanBase Reader by using the code editor

In the following code, a synchronization node is configured to read data from ApsaraDB for OceanBase. For more information, see Create a synchronization node by using the code editor.

```
{
    "type": "job",
    "steps": [
        {
            "stepType": "apsaradb_for_OceanBase", // The reader type.
            "parameter": {
                "datasource": "", // The name of the data source.
                "where": "",
                "column": [ // The names of the columns from which you want to read data.
                    "id",
                    "name"
                ],
                "splitPk": ""
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "stream",
            "parameter": {
                "print": false,
                "fieldDelimiter": ","
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0" // The maximum number of dirty data records allowed.
        },
        "speed": {
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent": 1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    }
}
```

5.2.34. Hologres Reader

Hologres Reader reads data from Hologres, converts the data to a format that is readable to Data Integration, and then sends the data to a writer. The writer writes the data to the related destination.

Background information

Notice

- Hologres Reader supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration, Use the default resource group, and Create a custom resource group for Data Integration.
- Batch synchronization nodes cannot be used to synchronize Hologres foreign tables.

Hologres Reader reads data from Hologres tables by using PostgreSQL statements. The number of parallel threads that are used to read data is based on the number of shards in the Hologres table from which you want to read data. One SELECT statement is executed for each shard.

- When you execute the `CREATE TABLE` statement to create a table in Hologres, you can use the `CALL set_table_property('table_name', 'shard_count', 'xx')` command to configure the number of shards for the table.

  By default, the shard_count field is set to the default number of table shards for your Hologres database. The configurations of your Hologres instance determine the default number of table shards for your Hologres database.
- A SELECT statement uses the shard that is specified by the built-in field hg_shard_id of the source Hologres table to query data.

Parameters

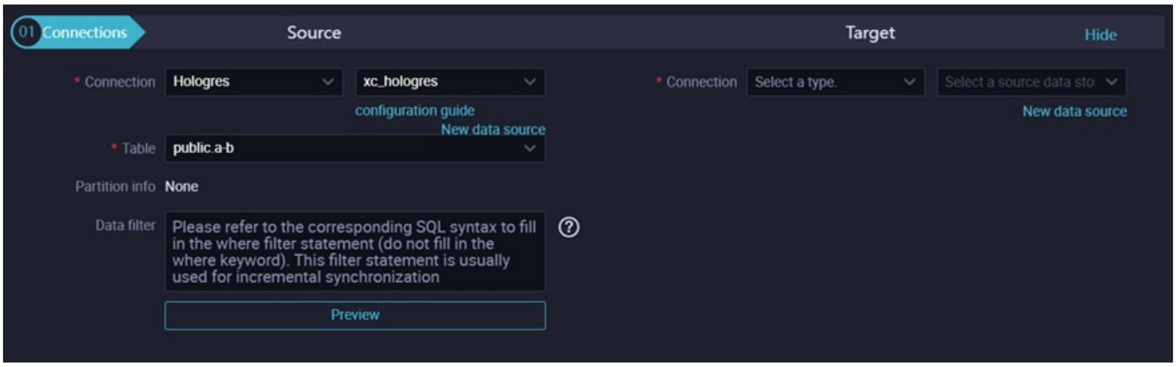
Parameter	Description	Required	Default value
endpoint	The endpoint of the source Hologres instance. Specify the value in the format of `instance-id-region-endpoint.hologres.aliyuncs.com:Port number`. You can view the endpoint of a Hologres instance on the configuration page of the instance in the Hologres console. The endpoint of a Hologres instance varies based on the network type, including the classic network, Internet, and virtual private cloud (VPC). Set this parameter based on the type of the network where your exclusive resource group for Data Integration and the Hologres instance reside. If an invalid endpoint is specified, the connection between the exclusive resource group for Data Integration and the Hologres instance may fail, or data synchronization performance may be poor. The endpoints for the three network types are in the following formats: • Classic network endpoint: `instance-id-region-endpoint-internal.hologres.aliyuncs.com:Port number` • Public endpoint: `instance-id-region-endpoint.hologres.aliyuncs.com:Port number` • VPC endpoint: `instance-id-region-endpoint-vpc.hologres.aliyuncs.com:Port number` We recommend that you deploy the exclusive resource group for Data Integration and the Hologres instance in the same zone of the same region. This helps ensure a successful network connection and the optimal data synchronization performance.	Yes	No default value
accessId	The AccessKey ID of the account that you use to connect to Hologres.	Yes	No default value
accessKey	The AccessKey secret of the account that you use to connect to Hologres. Make sure that the account is authorized to read data from the source table.	Yes	No default value
database	The name of the source database in the Hologres instance.	Yes	No default value

Parameter	Description	Required	Default value
table	The name of the table from which you want to read data. If the table is a partitioned table, specify the name of the table instead of the name of the partition from which you want to read data. Note Hologres Reader cannot read data from views.	Yes	No default value
column	The names of the columns from which you want to read data. The names of the primary key columns in the source table must be included. If you want to read data from all columns in the source table, set this parameter to `["*"]` .	Yes	No default value
partition	The partition key column and the related value of the source table, in the format of `column=value` . This parameter is valid only for partitioned tables. Notice • Hologres supports only list partitioning, and you can specify only one column as the partition key column. The data type of the partition key column must be INT4 or TEXT. • The value of this parameter must match the partition expression in the data definition language (DDL) statements that are used to create the source table. • You must specify a partition that exists and contains data.	No	Left empty, indicating that the source table is a non-partitioned table
fetchSize	The maximum number of data records to read from the source table at a time by using the SELECT statement.	No	*1,000*

Configure Hologres Reader by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



Source

* Connection: Hologres xc_hologres

* Table: public.a-b

Partition info: None

Data filter: Please refer to the corresponding SQL syntax to fill in the where filter statement (do not fill in the where keyword). This filter statement is usually used for incremental synchronization

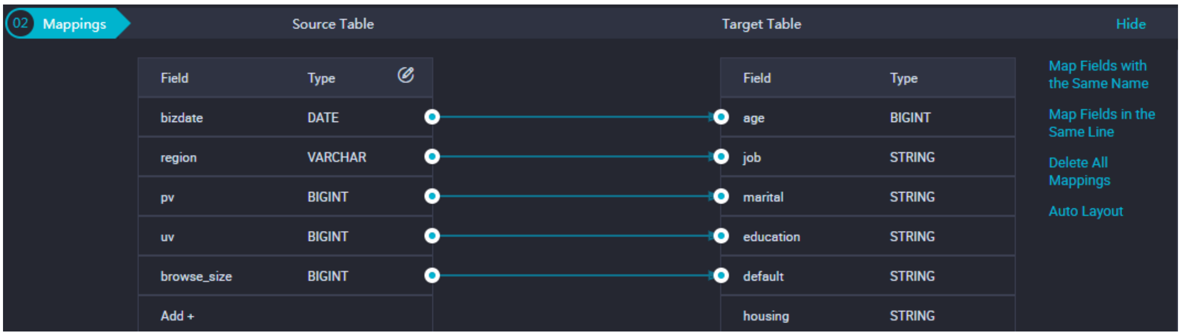
Target

* Connection: Select a type. Select a source data sto

Parameter	Description
Connection	The name of the data source from which you want to read data.
Table	The name of the table from which you want to read data. This parameter is equivalent to the table parameter that is specified in the preceding section.
Filter	The condition that is used to filter the data you want to read. The WHERE clause is not supported. The SQL syntax is determined by the selected data source.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.

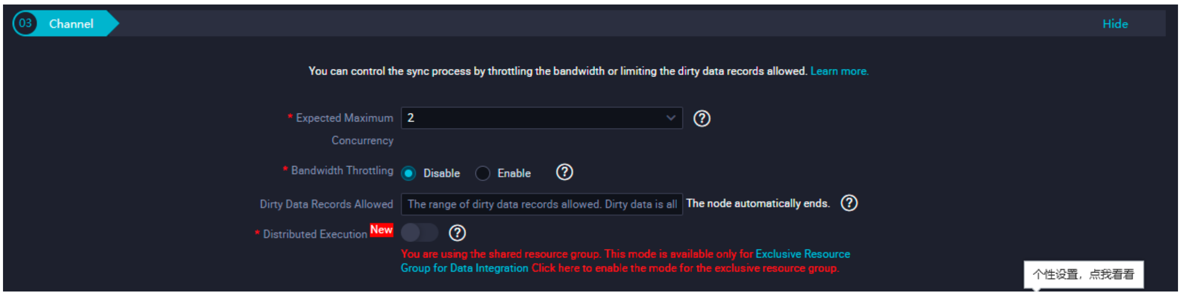


Field	Type
bizdate	DATE
region	VARCHAR
pv	BIGINT
uv	BIGINT
browse_size	BIGINT

Field	Type
age	BIGINT
job	STRING
marital	STRING
education	STRING
default	STRING
housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system sorts the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	Click Add to add a field. Take note of the following rules when you add a field: ○ You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ○ You can use scheduling parameters, such as ${bizdate}. ○ You can enter functions that are supported by relational databases, such as now() and count(1). ○ If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to avoid heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure Hologres Reader by using the code editor

- Read data from a non-partitioned Hologres table
  - In the following code, a synchronization node is configured to read data from a non-partitioned Hologres table:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"holo",// The reader type.
            "parameter":{
                "endpoint": "instance-id-region-endpoint.hologres.aliyuncs.com:port",
                "accessId": "***************", // The AccessKey ID of the account that you use to connect
to Hologres.
                "accessKey": "*******************", // The AccessKey secret of the account that you use to
connect to Hologres.
                "database": "postgres",
                "table": "holo_reader_****",
                "column" : [ // The names of the columns from which you want to read data.
                    "tag",
                    "id",
                    "title"
                ]
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates
that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled.
The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1,// The maximum number of parallel threads.
                  "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

- The following sample DDL statements are used to create a non-partitioned Hologres table:

  ```
  begin;
  drop table if exists holo_reader_basic_src;
  create table holo_reader_basic_src(
    tag text not null,
    id int not null,
    title text not null,
    body text,
    primary key (tag, id));
    call set_table_property('holo_reader_basic_src', 'orientation', 'column');
    call set_table_property('holo_reader_basic_src', 'shard_count', '3');
  commit;
  ```

- Read data from a partition in a partitioned Hologres table
  - In the following code, a synchronization node is configured to read data from a partition in a partitioned Hologres table in SDK mode.

    Note Exercise caution when you set the partition parameter.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"holo",// The reader type.
            "parameter":{
                "endpoint": "instance-id-region-endpoint.hologres.aliyuncs.com:port",
                "accessId": "***************", // The AccessKey ID of the account that you use to connect
to Hologres.
                "accessKey": "*******************", // The AccessKey secret of the account that you use to
connect to Hologres.
                "database": "postgres",
                "table": "holo_reader_basic_****",
                "partition": "tag=foo",
                "column" : [
                    "*"
                ],
                "fetchSize": "100"
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates
that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled.
The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1,// The maximum number of parallel threads.
                  "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

- The following sample DDL statements are used to create a partitioned Hologres table:

```
begin;
drop table if exists holo_reader_basic_part_src;
create table holo_reader_basic_part_src(
  tag text not null,
  id int not null,
  title text not null,
  body text,
  primary key (tag, id))
  partition by list( tag );
  call set_table_property('holo_reader_basic_part_src', 'orientation', 'column');
  call set_table_property('holo_reader_basic_part_src', 'shard_count', '3');
commit;
create table holo_reader_basic_part_src_1583161774228 partition of holo_reader_basic_part_src for values i
n ('foo');
# Make sure that the partition from which you want to read data is created and data is inserted into the p
artition.
postgres=# \d+ holo_reader_basic_part_src
                         Table "public.holo_reader_basic_part_src"
 Column |  Type   | Collation | Nullable | Default | Storage  | Stats target | Description
--------+---------+-----------+----------+---------+----------+--------------+-------------
 tag    | text    |           | not null |         | extended |              |
 id     | integer |           | not null |         | plain    |              |
 title  | text    |           | not null |         | extended |              |
 body   | text    |           |          |         | extended |              |
Partition key: LIST (tag)
Indexes:
    "holo_reader_basic_part_src_pkey" PRIMARY KEY, btree (tag, id)
Partitions: holo_reader_basic_part_src_1583161774228 FOR VALUES IN ('foo')
```

5.2.35. GDB Reader

This topic describes the parameters that are supported by Graph Database (GDB) Reader and how to configure GDB Reader by using the codeless user interface (UI) and code editor.

Background information

GDB is a real-time and reliable online database service that supports the Property Graph model and allows you to query and store highly connected datasets. GDB uses Apache TinkerPop Gremlin as the query language, which allows you to build queries that navigate highly connected datasets with improved efficiency.

> **Note**
>
> - Before you configure GDB Reader, configure a data source. For more information, see Configure a GDB data source.
> - You must separately configure data integration tasks of a vertex and an edge because the settings of the two tasks are different.

Limits

- You must configure two synchronization nodes to synchronize data about vertices and edges separately.
- The vertices or edges whose data is to be synchronized must have names for DataWorks to traverse and obtain related data.
- The primary key values of vertices and edges in GDB are of the STRING type. The type of data to be synchronized must be configured as the STRING type. If the configured type is a numeric type, such as LONG, GDB Reader forcibly converts the primary key values to the STRING type. If the conversion fails, the primary key values are lost.
- For the values of vertex or edge properties, the data type for the property values to be synchronized must be the same as the original data type in a GDB instance. If the data type for the property values is different from the original data type, GDB Reader forcibly converts the property values to the specified data type. If the conversion fails, the property values are lost.
- If you run a node to synchronize the vertex data multiple times, the obtained values of the SET property may be different.

- If you configure all properties in the JSON format, the SET property that contains only one value is regarded as a common property.
- Unless otherwise specified, field names or enumerated values in this topic are case-sensitive.
- GDB Reader supports only UTF-8 encoding. The synchronized data must be encoded in UTF-8.
- Only GDB 1.0.20 or later supports the SET property. Confirm the GDB version before you use the SET property.

Parameters

Parameter	Description	Required	Default value
host	The endpoint that is used to connect to the GDB instance. You can log on to the GDB console, find the instance that you want to configure, and click **View Instance Details** in the Actions column to view **Intranet URL**.	Yes	No default value
port	The port number that is used to connect to the GDB instance.	Yes	*8182*
username	The username that is used to connect to the GDB instance.	Yes	No default value
password	The password that is used to connect to the GDB instance.	Yes	No default value
labels	The label, which is the name of the vertex or edge. GDB Reader can read data from multiple vertices or edges at a time. In this case, the value of this parameter is an array, such as ["label1", "label2"].	Yes	No default value
labelType	The type of the label. Valid values: • VERTEX: a vertex. • EDGE: an edge.	Yes	No default value
column	The vertices or edges to be synchronized.	Yes	No default value
column -> name	The name of the vertex or edge property to be synchronized. This parameter is required if vertex or edge properties are to be synchronized.	Yes	No default value
column -> type	The data type for storing the vertex or edge property to be synchronized. • The primary key and label can only be of the STRING type. If you do not set the data type to STRING, data conversion fails. • Other properties can be of the INT, LONG, FLOAT, DOUBLE, BOOLEAN, or STRING type. • GDB Reader forcibly converts the obtained data to the specified type. If the conversion fails, the data record is lost.	Yes	No default value

<table>
<tr><th>Parameter</th><th>Description</th><th>Required</th><th>Default value</th></tr>
<tr><td>column -> columnType</td><td>

The category of the vertex or edge property to be synchronized.

- For both vertices and edges:
  - primaryKey: the primary key.
  - primaryLabel: the label.
- For vertices:
  - vertexProperty: a common property of the vertex.
  - vertexJsonProperty: a collection of the properties of the vertex, in the JSON format. If you set the columnType parameter to vertexJsonProperty, all properties are listed in this column. Other columns cannot contain the property of the vertex.

    Example of vertexJsonProperty:

    ```
    {
        "properties":[
            {"k":"name","t":"string","v":"tom","c":"set"},

    {"k":"name","t":"string","v":"jack","c":"set"},

    {"k":"gender","t":"string","v":"male","c":"single"}
        ]
    }
    ```

    The preceding code contains a multi-value property name and a single-value property gender. The name property has two records. Although the gender property is a multi-value property, it is regarded as a single-value property in this example because only one related record exists.
- For edges:
  - srcPrimaryKey: the primary key of the start vertex.
  - padstPrimaryKey: the primary key of the end vertex.
  - srcPrimaryLabel: the label of the start vertex.
  - dstPrimaryLabel: the label of the end vertex.
  - edgeProperty: a property of the edge.
  - edgeJsonProperty: a collection of the properties of the edge, in the JSON format. If you set the columnType parameter to edgeJsonProperty, all properties are listed in this column. Other columns cannot contain the property of the edge.

    Example of edgeJsonProperty:

    ```
    {
        "properties":[
            {"k":"name","t":"string","v":"tom"},
            {"k":"gender","t":"string","v":"male"}
    ]
    }
    ```

    An edge does not support multi-value properties or the c field.

</td><td>Yes</td><td>No default value</td></tr>
</table>

Configure GDB Reader by using the codeless UI

This method is not supported.

Configure GDB Reader by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, two synchronization nodes are configured to read data from a GDB instance. For more information about the parameters, see Parameters.

- Configure a synchronization node to read data about vertices from a GDB instance

```
{
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    },
    "setting":{
        "errorLimit":{
            "record":"100"  // The maximum number of dirty data records allowed.
        },
        "jvmOption":"",
        "speed":{
            "concurrent":3,
            "throttle":true,/// Specifies whether to enable bandwidth throttling. The value false indicates t
hat bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The
mbps parameter takes effect only when the throttle parameter is set to true.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "steps":[
        {
            "category":"reader",
            "name":"Reader",
            "parameter":{
                "host": "gdb-xxxxxx.aliyuncs.com", // The endpoint that is used to connect to the GDB instanc
e.
                "port": 8182, // The port number that is used to connect to the GDB instance.
                "username": "gdb", // The username that is used to connect to the GDB instance.
                "password": "gdb", // The password that is used to connect to the GDB instance.
                "labelType": "VERTEX", // The label type.
                "labels": ["label1", "label2"],  // The labels of the vertices to be synchronized. If this pa
rameter is left empty, all vertices are synchronized.
                "column": [
                    {
                        "name": "id",               // The name of the vertex property.
                        "type": "string",           // The data type for storing the data to be synchronized.

                        "columnType": "primaryKey"  // The category of the vertex property. The value primary
Key indicates that the synchronized data is the primary key of the vertex and is of the STRING type in the GD
B instance.
                    },
                    {
                        "name": "label",              // The name of the vertex property.
                        "type": "string",             // The data type for storing the data to be synchronize
d.
                        "columnType": "primaryLabel"  // The category of the vertex property. The value prima
ryLabel indicates that the synchronized data is the label of the vertex and is of the STRING type in the GDB
instance.
                    },
                    {
                        "name": "age",                   // The name of the vertex property.
                        "type": "int",                   // The data type for storing the data to be synchron
ized.
                        "columnType": "vertexProperty"   // The category of the vertex property. The value ve
rtexProperty indicates a common vertex property.
```

```
                    }
                ]
            },
            "stepType":"gdb"
        },
        {
            "category":"writer",
            "name":"Writer",
            "parameter":{
                "print": true
            },
            "stepType":"stream"
        }
    ]
}
```

- Configure a synchronization node to read data about edges from a GDB instance

```
{
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    },
    "setting":{
        "errorLimit":{
            "record":"100"  // The maximum number of dirty data records allowed.
        },
        "jvmOption":"",
        "speed":{
            "concurrent":3,
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates th
at bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The m
bps parameter takes effect only when the throttle parameter is set to true.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "steps":[
        {
            "category":"reader",
            "name":"Reader",
            "parameter":{
                "host": "gdb-xxxxxx.aliyuncs.com", // The endpoint that is used to connect to the GDB instanc
e.
                "port": 8182, // The port number that is used to connect to the GDB instance.
                "username": "gdb", // The username that is used to connect to the GDB instance.
                "password": "gdb", // The password that is used to connect to the GDB instance.
                "labelType": "EDGE", // The label type.
                "labels": ["label1", "label2"],  // The labels of the edges to be synchronized. If this param
eter is left empty, all edges are synchronized.
                "column": [
                    {
                        "name": "id",               // The name of the edge property.
                        "type": "string",           // The data type for storing the data to be synchronized.

                        "columnType": "primaryKey"  // The category of the edge property. The value primaryKe
y indicates that the synchronized data is the primary key of the edge and is of the STRING type in the GDB in
stance.
                    },
                    {
                        "name": "label",              // The name of the edge property.
                        "type": "string",             // The data type for storing the data to be synchronize
d.
```

```
                        "columnType": "primaryLabel"  // The category of the edge property. The value primary
Label indicates that the synchronized data is the label of the edge and is of the STRING type in the GDB inst
ance.
                    },
                    {
                        "name": "srcId",               // The name of the edge property.
                        "type": "string",              // The data type for storing the data to be synchroniz
ed.
                        "columnType": "srcPrimaryKey"  // The category of the edge property. The value srcPri
maryKey indicates that the synchronized data is the primary key of the start vertex and is of the STRING type
in the GDB instance.
                    },
                    {
                        "name": "srcLabel",               // The name of the edge property.
                        "type": "string",                 // The data type for storing the data to be synchro
nized.
                        "columnType": "srcPrimaryLabel"   // The category of the edge property. The value src
PrimaryLabel indicates that the synchronized data is the label of the start vertex and is of the STRING type
in the GDB instance.
                    },
                    {
                        "name": "dstId",                    // The name of the edge property.
                        "type": "string",                   // The data type for storing the data to be synch
ronized.
                        "columnType": "dstPrimaryKey"       // The category of the edge property. The value d
stPrimaryKey indicates that the synchronized data is the primary key of the end vertex and is of the STRING t
ype in the GDB instance.
                    },
                    {
                        "name": "dstLabel",                 // The name of the edge property.
                        "type": "string",                   // The data type for storing the data to be synch
ronized.
                        "columnType": "dstPrimaryLabel"     // The category of the edge property. The value d
stPrimaryLabel indicates that the synchronized data is the label of the end vertex and is of the STRING type
in the GDB instance.
                    },
                    {
                        "name": "weight",               // The name of the edge property.
                        "type": "double",               // The data type for storing the data to be synchroni
zed.
                        "columnType": "edgeProperty"    // The category of the edge property. The value edgeP
roperty indicates a common edge property.
                    }
                ]
            },
            "stepType":"gdb"
        },
        {
            "category":"writer",
            "name":"Writer",
            "parameter":{
                "print": true
            },
            "stepType":"stream"
        }
    ]
}
```

5.2.36. RestAPI Reader

This topic describes the data types and parameters that are supported by RestAPI Reader and how to configure RestAPI Reader by using the codeless user interface (UI) and code editor. Before you create a Data Integration node, you can read this topic to familiarize yourself with the data types and parameters that you must set for RestAPI Reader.

Context

RestAPI Reader reads data from the responses returned by RESTful APIs, converts the data into that of the data types supported by Data Integration, and then sends the converted data to a writer. RestAPI Reader can read data of the following data types from the JSON-formatted responses returned by RESTful APIs: INT, BOOLEAN, DATE, DOUBLE, FLOAT, LONG, and STRING.

Data types

Category	Data Integration data type
Integer	LONG and INT
String	STRING
Floating point	DOUBLE and FLOAT
Boolean value	BOOLEAN
Date and time	DATE

Parameters

Before you perform data integration, you must add a data source and configure it as the source or destination. You must also configure the data that you want to integrate and the data types. During data integration, RestAPI Reader reads data from the source, and a writer writes data to the destination.

The following table describes the parameters that you can set when you use RestAPI Reader to read data from a RestAPI data source.

> **Note** You can set the parameters that are described in the following table when you add a RestAPI data source and configure a Data Integration node.
>
> You cannot set scheduling parameters for RestAPI Reader.

Parameter	Description	Required	Default value
url	The URL of the RESTful API.	Yes	No default value
dataMode	The method that RestAPI Reader uses to read data from the JSON-formatted response returned by the RESTful API. Valid values: • oneData: RestAPI Reader extracts one data record. • multiData: RestAPI Reader extracts a JSON array and transfers multiple data records to a writer.	Yes	No default value
responseType	The format of the response returned by the RESTful API. Only the JSON format is supported.	Yes	JSON
column	The names of the fields from which you want to read data. The type parameter specifies the data type of a field. The name parameter specifies the JSON-formatted path where the field is located. You can set the column parameter in the following format: "column":[{"type":"long","name":"a.b" // Query data in the a.b path.},{"type":"string","name":"a.c"// Query data in the a.c path.}] You must specify the type and name parameters for each field.	Yes	No default value
dataPath	The path in which the JSON-formatted data record or JSON array is queried.	No	No default value

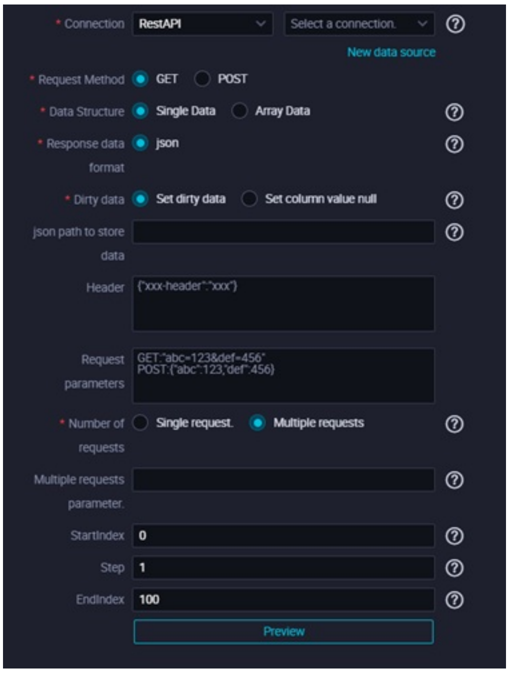
Parameter	Description	Required	Default value
method	The request method. Valid values: get and post.	Yes	No default value
customHeader	The header information transferred to the RESTful API.	No	No default value
parameters	The parameter information transferred to the RESTful API. • If the method parameter is set to get, set this parameter to `abc=1&def=1`. • If the method parameter is set to post, specify JSON parameters.	No	No default value
dirtyData	The processing mechanism to take effect when no data is found in the specified JSON-formatted path of the column parameter. Valid values: • dirty: If a specific data record cannot be found in the specified JSON-formatted path, this data record is marked as a dirty data record. • null: If a specific data record cannot be found in the specified JSON-formatted path, the column parameter is set to null.	Yes	dirty
requestTimes	The number of times RestAPI Reader requests to read data from the response returned by the RESTful API. Valid values: • single: only once • multiple: multiple times	Yes	single
requestParam	If the requestTimes parameter is set to multiple, you must specify a parameter in each repeatedly transmitted request. For example, if you specify the pageNumber parameter, RestAPI Reader passes the pageNumber parameter to the RESTful API based on the settings of the startIndex, endIndex, and step parameters.	No	No default value
startIndex	The start point of requests. The data at the start point is also requested.	No	No default value
endIndex	The end point of requests. The data at the end point is also requested.	No	No default value
step	The step at which requests are sent.	No	No default value

Parameter	Description	Required	Default value
authType	The authentication method. Valid values: - Basic Auth: basic authentication If the data source supports username- and password-based authentication, you can select Basic Auth and configure the username and password that can be used for authentication. During data integration, the username and password are transferred to the RESTful API URL for authentication. The data source is connected only after the authentication is successful. - Token Auth: token-based authentication If the data source supports token-based authentication, you can select Token Auth and configure a fixed token value that can be used for authentication. During data integration, the token is contained in the request header, such as {"Authorization":"Bearer TokenXXXXXX"}, and transferred to the RESTful API URL for authentication. The data source is connected only after the authentication is successful. - Aliyun API Signature: Alibaba Cloud API signature-based authentication If the following conditions are met, you can select Aliyun API Signature and configure the AccessKey ID and AccessKey secret that can be used for authentication: The data source that you want to connect is an Alibaba Cloud service, and the API of this service supports AccessKey pair-based authentication.	No	No default value
authUsername/authPassword	The username and password used for basic authentication.	No	No default value
authToken	The token used for token-based authentication.	No	No default value
accessKey/accessSecret	The AccessKey pair used for Alibaba Cloud API signature-based authentication.	No	No default value

Configure RestAPI Reader by using the codeless UI

1. Configure data sources.

Configure **Source** and Target for the synchronization node.



* Connection: RestAPI | Select a connection. ⓘ

* Request Method: ● GET ○ POST

* Data Structure: ● Single Data ○ Array Data

* Response data format: ● json

* Dirty data: ● Set dirty data ○ Set column value null

json path to store data:

Header: {"xxx-header":"xxx"}

Request parameters: GET:"abc=123&def=456"
POST:{"abc":123,"def":456}

* Number of requests: ○ Single request. ● Multiple requests

Multiple requests parameter.:

StartIndex: 0

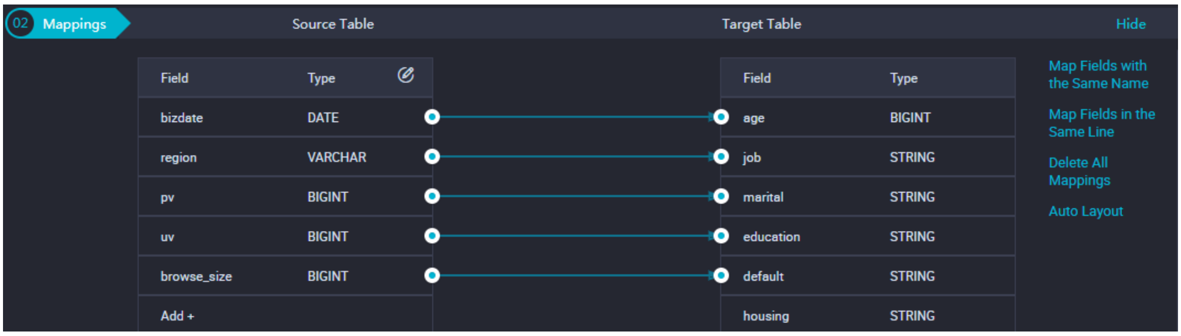
Step: 1

EndIndex: 100

Parameter	Description
Connection	The name of the data source from which you want to read data. Select **RestAPI** from the left-side drop-down list and the name of a data source that you configured from the right-side drop-down list.
Request Method	This parameter is equivalent to the method parameter that is described in the preceding section.
Data Structure	This parameter is equivalent to the dataMode parameter that is described in the preceding section.
json path to store data	This parameter is equivalent to the dataPath parameter that is described in the preceding section.
Response data format	This parameter is equivalent to the responseType parameter that is described in the preceding section. Only the JSON format is supported.
Dirty data	This parameter is equivalent to the dirtyData parameter that is described in the preceding section. Default value: Set dirty data.
Header	This parameter is equivalent to the customHeader parameter that is described in the preceding section.
Request parameters	This parameter is equivalent to the parameters parameter that is described in the preceding section.
Number of requests	This parameter is equivalent to the requestTimes parameter that is described in the preceding section.
StartIndex/Step/EndIndex	These parameters are equivalent to the startIndex, step, and endIndex parameters that are described in the preceding section.

2. Configure field mappings.

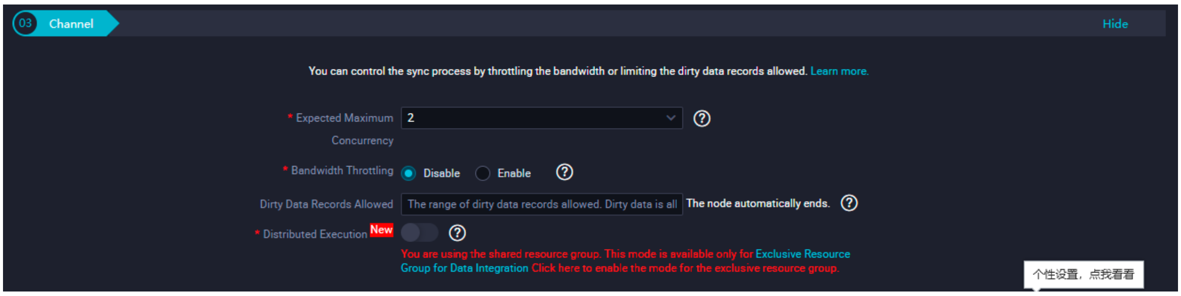
Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



Field	Type	Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
Add +		housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout to sort the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	Click **Add** to add a field. Take note of the following rules when you add a field: ◦ You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ◦ You can use scheduling parameters, such as ${bizdate}. ◦ You can enter functions that are supported by relational databases, such as now() and count(1). ◦ If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of concurrent threads that the synchronization node can use to read data from the source or write data to the destination. You can configure the concurrency for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.

Parameter	Description
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	This parameter is not supported for synchronization nodes that use RestAPI Reader.

Configure RestAPI Reader by using the code editor

In the following sample code, a synchronization node is configured to read data from the response returned by a RESTful API:

```
{
    "type":"job",
    "version":"2.0",
    "steps":[
        {
            "stepType":"restapi",
            "parameter":{
                "url":"http://127.0.0.1:5000/get_array5",
                "dataMode":"oneData",
                "responseType":"json",
                "column":[
                    {
                        "type":"long",
                        "name":"a.b"  // Query data in the a.b path.
                    },
                    {
                        "type":"string",  // Query data in the a.c path.
                        "name":"a.c"
                    }
                ],
                "dirtyData":"null",
                "method":"get",
                "defaultHeader":{
                    "X-Custom-Header":"test header"
                },
                "customHeader":{
                    "X-Custom-Header2":"test header2"
                },
                "parameters":"abc=1&amp;def=1"
            },
            "name":"restapireader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":""
        },
        "speed":{
            "throttle":true,  // Specifies whether to enable bandwidth throttling. A value of false specifies that bandwidth throttling is disabled, and a value of true specifies that bandwidth throttling is enabled. The mbps parameter takes effect only if the throttle parameter is set to true.
            "concurrent":1,  // The maximum number of concurrent threads allowed.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

Take note of the following information when you configure RestAPI Reader by using the code editor:

```
After RestAPI Reader sends an HTTP or HTTPS request, a JSON-formatted response is returned. The dataPath parame
ter is used to specify the path in which the JSON-formatted data record or JSON array is queried. The following
part provides two sample responses:
In the following sample response, a JSON array is returned for the DATA parameter that contains the business da
ta.
{
    "HEADER": {
        "BUSID": "bid1",
        "RECID": "uuid",
        "SENDER": "dc",
        "RECEIVER": "pre",
        "DTSEND": "202201250000"
    },
    "DATA": [
        {
            "SERNR": "sernr1"
        },
        {
            "SERNR": "sernr2"
        }
    ]
}
To extract multiple data records from the JSON array and transfer the data records to a writer, you must specif
y the column parameter in the "column": [ "SERNR" ] format, the dataMode parameter in the "dataMode": "multiDat
a" format, and the dataPath parameter in the "dataPath": "DATA" format.
In the following sample response, a JSON object is returned for the content.DATA parameter that contains the bu
siness data.
{
    "HEADER": {
        "BUSID": "bid1",
        "RECID": "uuid",
        "SENDER": "dc",
        "RECEIVER": "pre",
        "DTSEND": "202201250000"
    },
    "content": {
        "DATA": {
            "SERNR": "sernr2"
        }
    }
}
To extract one data record from the JSON object and transfer the data record to a writer, you must specify the
column parameter in the "column": [ "SERNR" ] format, the dataMode parameter in the "dataMode": "oneData" forma
t, and the dataPath parameter in the "dataPath": "content.DATA" format.
```

5.2.37. SAP HANA Reader

This topic describes the data types and parameters that are supported by SAP HANA Reader and how to configure SAP HANA Reader by using the codeless user interface (UI) and code editor. Before you create a Data Integration node, you can refer to this topic to familiarize yourself with the data types and parameters that you must configure for SAP HANA Reader to read data from data sources.

Context

SAP HANA Reader connects to a remote SAP HANA database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and sends the statement to the database. The system executes the statement on the database and returns data. Then, SAP HANA Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

Data types

The following table lists the data types that are supported by SAP HANA Reader.

Category	SAP HANA data type
Integer	INT, TINYINT, SMALLINT, MEDIUMINT, and BIGINT
Floating point	FLOAT, DOUBLE, and DECIMAL
String	VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT
Date and time	DATE, DATETIME, TIMESTAMP, TIME, and YEAR
Boolean	BIT and BOOL
Binary	TINYBLOB, MEDIUMBLOB, BLOB, LONGBLOB, and VARBINARY

Notice

- Data types that are not listed in the preceding table are not supported.
- SAP HANA Reader processes TINYINT(1) as an integer data type.

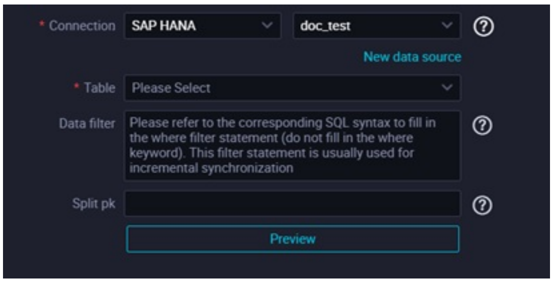
Parameters

Parameter	Description
username	The username that is used to log on to the SAP HANA database.
password	The password that is used to log on to the SAP HANA database.
column	The names of the columns from which you want to read data. To read data from all the columns in the source table, set this parameter to an asterisk (*).
table	The name of the table from which you want to read data.
jdbcUrl	The JDBC URL that is used to connect to SAP HANA. Example: jdbc:sap://127.0.0.1:30215?currentschema=TEST.
splitPk	The field that is used for data sharding when SAP HANA Reader reads data. If you specify this parameter, the source table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. You can specify a field of an integer data type for the splitPk parameter. If the source table does not contain fields of integer data types, you can leave this parameter empty.

Configure SAP HANA Reader by using the codeless UI

1. Configure data sources.

   Configure **Source** and Target for the synchronization node.



* Connection: SAP HANA | doc_test

* Table: Please Select

Data filter: Please refer to the corresponding SQL syntax to fill in the where filter statement (do not fill in the where keyword). This filter statement is usually used for incremental synchronization

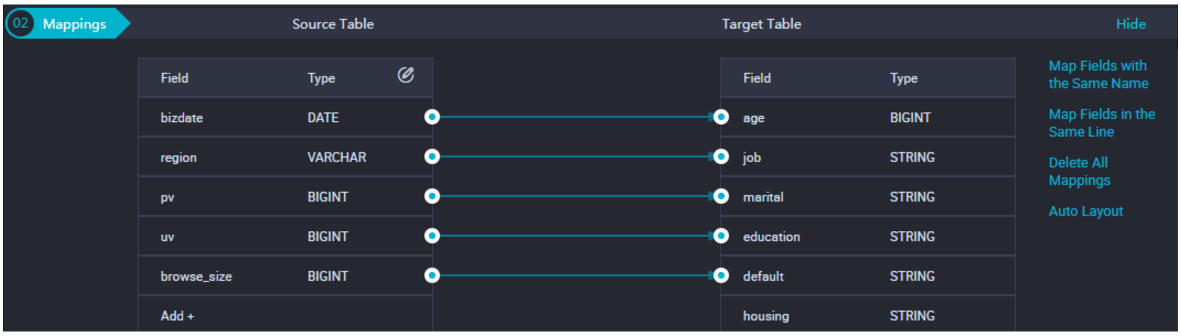
Split pk:

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.

Parameter	Description
Table	This parameter is equivalent to the table parameter that is described in the preceding section.
Filter	The condition that is used to filter the data you want to read. Filtering based on the LIMIT keyword is not supported. The SQL syntax is determined by the selected data source.
Shard Key	The shard key. You can use a column in the source table as the shard key. We recommend that you use the primary key column or an indexed column. Only integer columns are supported. If you set this parameter, data sharding is performed based on the value of this parameter, and parallel threads can be used to read data. This improves data synchronization efficiency. **Note** The Shard Key parameter is displayed only after you select the data source for the synchronization node.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

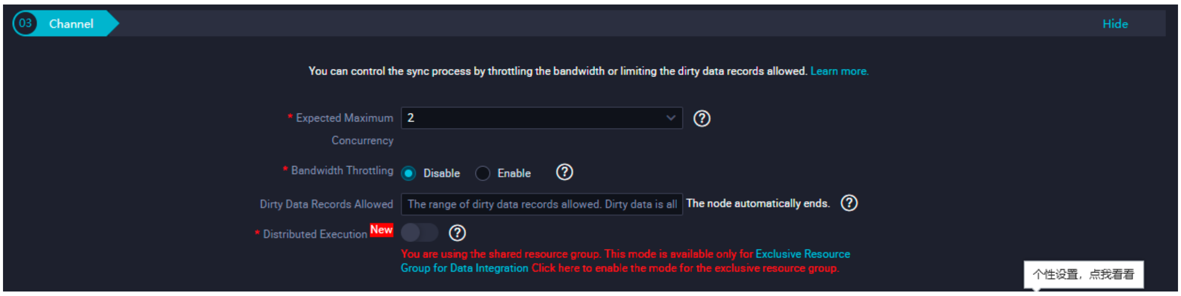
   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



Field	Type	Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
Add +		housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	Click **Add** to add a field. Take note of the following rules when you add a field: ◦ You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ◦ You can use scheduling parameters, such as ${bizdate}. ◦ You can enter functions that are supported by relational databases, such as now() and count(1). ◦ If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of concurrent threads that the synchronization node can use to read data from the source or write data to the destination. You can configure the concurrency for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	This parameter is not supported for synchronization nodes that use RestAPI Reader.

Configure SAP HANA Reader by using the code editor

The following sample code provides examples on how to configure a synchronization node to read data from a table that is not sharded and how to configure a synchronization node to read data from a sharded table.

- Configure a synchronization node to read data from a table that is not sharded

```
{
    "type":"job",
    "version":"2.0", // The version number.
    "steps":[
        {
            "stepType":"saphana",// The reader type.
            "parameter":{
                "column":[ // The names of the columns from which you want to read data.
                    "id"
                ],
                "connection":[
                    {   "querySql":["select a,b from join1 c join join2 d on c.id = d.id;"], // The SQL state
ment that is used to read data from the source table.
                        "datasource":"", // The name of the data source.
                        "table":[// The name of the source table. The table name must be enclosed in brackets
[].
                            "xxx"
                        ]
                    }
                ],
                "where":"",// The WHERE clause.
                "splitPk":"", // The shard key.
                "encoding":"UTF-8"// The encoding format.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. A value of false indicates t
hat bandwidth throttling is disabled, and a value of true indicates that bandwidth throttling is enabled. The
mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

- Configure a synchronization node to read data from a sharded table

> **Note** When you configure a synchronization node to read data from a sharded SAP HANA table, you can select multiple partitions that have the same schema.

```
{
    "type": "job",
    "version": "1.0",
    "configuration": {
        "reader": {
            "plugin": "saphana",
            "parameter": {
                "connection": [
                    {
                        "table": [
                            "tbl1",
                            "tbl2",
                            "tbl3"
                        ],
                        "datasource": "datasourceName1"
                    },
                    {
                        "table": [
                            "tbl4",
                            "tbl5",
                            "tbl6"
                        ],
                        "datasource": "datasourceName2"
                    }
                ],
                "singleOrMulti": "multi",
                "splitPk": "db_id",
                "column": [
                    "id", "name", "age"
                ],
                "where": "1 < id and id < 100"
            }
        },
        "writer": {
        }
    }
}
```

5.2.38. KingbaseES Reader

This topic describes the data types and parameters that are supported by KingbaseES Reader and how to configure KingbaseES Reader by using the codeless user interface (UI) and code editor. Before you create a Data Integration node, you can refer to this topic to familiarize yourself with the data types and parameters that you must configure for KingbaseES Reader to read data from data sources.

Context

KingbaseES Reader connects to a remote KingbaseES database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the statement on the database and returns data. Then, KingbaseES Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

Data types

The following table lists the data types that are supported by KingbaseES Reader.

Category	SAP KingbaseES data type
Integer	INT, TINYINT, SMALLINT, MEDIUMINT, and BIGINT
Floating point	FLOAT, DOUBLE, and DECIMAL
String	VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT

Category	SAP KingbaseES data type
Date and time	DATE, DATETIME, TIMESTAMP, TIME, and YEAR
Boolean	BIT and BOOLEAN
Binary	TINYBLOB, MEDIUMBLOB, BLOB, LONGBLOB, and VARBINARY

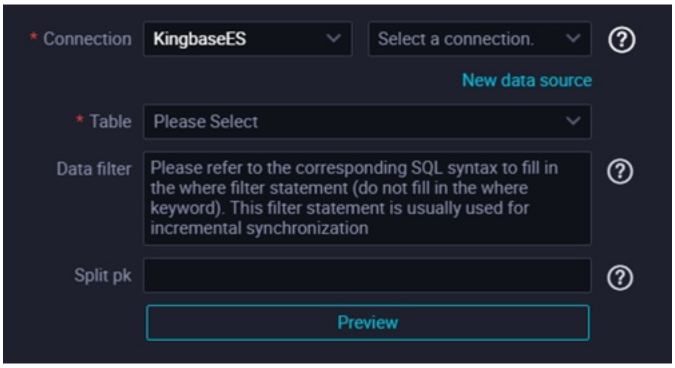
Notice

- Data types that are not listed in the preceding table are not supported.
- KingbaseES Reader processes TINYINT(1) as an integer data type.

Configure KingbaseES Reader by using the codeless UI

1. Configure data sources.

   Configure **Source** and Target for the synchronization node.



* Connection: KingbaseES · Select a connection.

New data source

* Table: Please Select

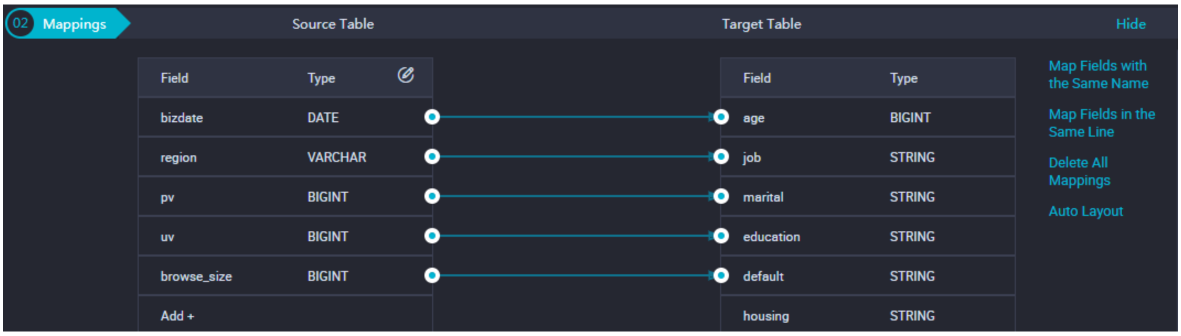
Data filter: Please refer to the corresponding SQL syntax to fill in the where filter statement (do not fill in the where keyword). This filter statement is usually used for incremental synchronization

Split pk:

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	This parameter is equivalent to the table parameter that is described in the preceding section.
Filter	The condition that is used to filter the data you want to read. Filtering based on the LIMIT keyword is not supported. The SQL syntax is determined by the selected data source.
Shard Key	The shard key. You can use a column in the source table as the shard key. We recommend that you use the primary key column or an indexed column. Only integer columns are supported. If you set this parameter, data sharding is performed based on the value of this parameter, and parallel threads can be used to read data. This improves data synchronization efficiency. **Note** The Shard Key parameter is displayed only after you select the data source for the synchronization node.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section.

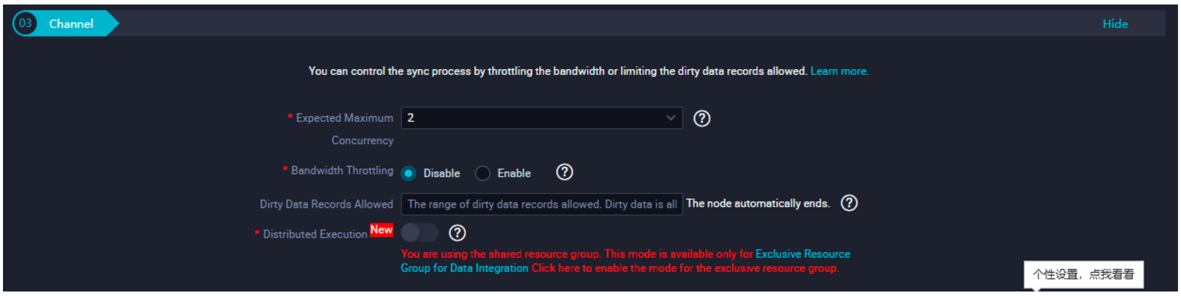
Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.



Field	Type	Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
Add +		housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	Click **Add** to add a field. Take note of the following rules when you add a field: ○ You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ○ You can use scheduling parameters, such as ${bizdate}. ○ You can enter functions that are supported by relational databases, such as now() and count(1). ○ If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of concurrent threads that the synchronization node can use to read data from the source or write data to the destination. You can configure the concurrency for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.

Parameter	Description
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	This parameter is not supported for synchronization nodes that use RestAPI Reader.

Configure KingbaseES Reader by using the code editor

The following sample code provides examples on how to configure a synchronization node to read data from a table that is not sharded and how to configure a synchronization node to read data from a sharded table.

- Configure a synchronization node to read data from a table that is not sharded

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"kingbasees",// The reader type.
            "parameter":{
                "column":[// The names of the columns from which you want to read data.
                    "id"
                ],
                "connection":[
                    {   "querySql":["select a,b from join1 c join join2 d on c.id = d.id;"], // The SQL state
ment that is used to read data from the source table.
                        "datasource":"",// The name of the data source.
                        "table":[// The name of the source table. The table name must be enclosed in brackets
[].
                            "xxx"
                        ]
                    }
                ],
                "where":"",// The WHERE clause.
                "splitPk":"",// The shard key.
                "encoding":"UTF-8"// The encoding format.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates th
at bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The m
bps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

- Configure a synchronization node to read data from a sharded table

> **Note** When you configure a synchronization node to read data from a sharded KingbaseES table, you can select multiple partitions that have the same schema.

```
{
    "type": "job",
    "version": "1.0",
    "configuration": {
        "reader": {
            "plugin": "kingbasees",
            "parameter": {
                "connection": [
                    {
                        "table": [
                            "tbl1",
                            "tbl2",
                            "tbl3"
                        ],
                        "datasource": "datasourceName1"
                    },
                    {
                        "table": [
                            "tbl4",
                            "tbl5",
                            "tbl6"
                        ],
                        "datasource": "datasourceName2"
                    }
                ],
                "singleOrMulti": "multi",
                "splitPk": "db_id",
                "column": [
                    "id", "name", "age"
                ],
                "where": "1 < id and id < 100"
            }
        },
        "writer": {
        }
    }
}
```

5.2.39. DM Reader

This topic describes the data types and parameters that are supported by DM Reader and how to configure DM Reader by using the codeless user interface (UI) and code editor.

Notice DM Reader supports only exclusive resource groups for Data Integration, but not the default resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.

Background information

DM Reader reads data from databases such as Dameng (DM), Db2, PPAS, and Sybase. DM Reader is commonly used to read data from relational databases. To enable DM Reader to read data from a relational database, you must register the driver for the relational database.

DM Reader connects to a remote database by using Java Database Connectivity (JDBC), generates an SQL statement based on your configurations, and then sends the statement to the database. The system executes the statement on the database and returns data. Then, DM Reader assembles the returned data into abstract datasets of the data types supported by Data Integration and sends the datasets to a writer.

- DM Reader generates the SQL statement based on the settings of the table, column, and where parameters and sends the generated statement to the remote database.
- If you specify the querySql parameter, DM Reader sends the value of this parameter to the remote database.

DM Reader supports most data types of common relational databases, such as numeric and string data types. Make sure that the data types of your database are supported.

Parameters

Parameter	Description	Required	Default value
jdbcUrl	The JDBC URL that you can use to connect to the source database. The format must be in accordance with the official specifications of the database. You can also specify the information of the attachment facility. The format varies based on the database type. Data Integration selects the most suitable driver based on the format and uses this driver to read data from the source database. - Format for DM databases: `jdbc:dm://IP address:Port number/database` - Format for Db2 databases: `jdbc:db2://IP address:Port number/database` - Format for PPAS databases: `jdbc:edb://IP address:Port number/database` You can use the following method to enable DM Reader to support a new type of database: - Go to the directory of DM Reader. In the directory, ${DATAX_HOME} indicates the main directory of Data Integration. - Open the *plugin.json* file in the directory of DM Reader and add the driver of the database to the drivers array in the file. When a synchronization node runs, DM Reader dynamically selects the most suitable database driver to connect to the database.	Yes	No default value

```
{
    "name": "rdbmsreader",
    "class": 
"com.alibaba.datax.plugin.reader.rdbmsreader.RdbmsReader",
    "description": "useScene: prod. mechanism: Jdbc connection using 
the database, execute select sql, retrieve data from the ResultSet. 
warn: The more you know about the database, the less problems you 
encounter.",
    "developer": "alibaba",
    "drivers": [
        "dm.jdbc.driver.DmDriver",
        "com.ibm.db2.jcc.DB2Driver",
        "com.sybase.jdbc3.jdbc.SybDriver",
        "com.edb.Driver"
    ]
}
```
- Add the package of the driver to the libs subdirectory in the directory of DB Reader.
```
$tree
.
|-- libs
|   |-- Dm7JdbcDriver16.jar
|   |-- commons-collections-3.0.jar
|   |-- commons-io-2.4.jar
|   |-- commons-lang3-3.3.2.jar
|   |-- commons-math3-3.1.1.jar
|   |-- datax-common-0.0.1-SNAPSHOT.jar
|   |-- datax-service-face-1.0.23-20160120.024328-1.jar
|   |-- db2jcc4.jar
|   |-- druid-1.0.15.jar
|   |-- edb-jdbc16.jar
|   |-- fastjson-1.1.46.sec01.jar
|   |-- guava-r05.jar
|   |-- hamcrest-core-1.3.jar
|   |-- jconn3-1.0.0-SNAPSHOT.jar
|   |-- logback-classic-1.0.13.jar
|   |-- logback-core-1.0.13.jar
|   |-- plugin-rdbms-util-0.0.1-SNAPSHOT.jar
|   `-- slf4j-api-1.7.10.jar
|-- plugin.json
|-- plugin_job_template.json
`-- rdbmsreader-0.0.1-SNAPSHOT.jar
```

Parameter	Description	Required	Default value
username	The username that is used to connect to the source database.	Yes	No default value
password	The password that is used to connect to the source database.	Yes	No default value
table	The name of the table from which you want to read data.	Yes	No default value

Parameter	Description	Required	Default value
column	The names of the columns from which you want to read data. Specify the names in a JSON array. The default value is [*], which indicates all the columns in the source table. - You can select specific columns to read. - The column order can be changed. This indicates that you can specify columns in an order different from the order specified by the schema of the source table. - Constants are supported. The column names must be arranged in the JSON format, such as `["id","1", "'bazhen.csy'", "null", "to_char(a + 1)", "2.3" , "true"]` . ○ id: a column name. ○ 1: an integer constant. ○ 'bazhen.csy': a string constant. ○ null: a null pointer. ○ to_char(a + 1): a function expression. ○ 2.3: a floating-point constant. ○ true: a Boolean value. - The column parameter must explicitly specify all the columns from which you want to read data. The parameter cannot be left empty.	Yes	No default value
splitPk	The field that is used for data sharding when DM Reader reads data. If you specify this parameter, the table is sharded based on the value of this parameter. Data Integration then runs parallel threads to read data. This way, data can be synchronized more efficiently. - We recommend that you set the splitPk parameter to the name of the primary key column of the table. Data can be evenly distributed to different shards based on the primary key column, instead of being intensively distributed only to specific shards. - The splitPk parameter supports sharding only for data of integer data types. If you set this parameter to a field of an unsupported data type, such as a string, floating point, or date data type, DM Reader returns an error. - If you do not specify the splitPk parameter, DM Reader uses a single thread to read all the data in the source table.	No	No default value
where	The WHERE clause. DM Reader generates an SQL statement based on the settings of the column, table, and where parameters and uses the generated statement to read data. For example, when you perform a test, you can set the where parameter to limit 10. To read the data that is generated on the current day, you can set the where parameter to `gmt_create > $bizdate` . - You can use the WHERE clause to read incremental data. - If the where parameter is not provided or is left empty, DM Reader reads all data.	No	No default value
querySql	The SQL statement that is used for refined data filtering. If you specify this parameter, data is filtered based only on the value of this parameter. For example, if you want to join multiple tables for data synchronization, set this parameter to `select a,b from table_a join table_b on table_a.id = table_b.id` . If you specify this parameter, DM Reader ignores the settings of the column, table, and where parameters.	No	No default value
fetchSize	The number of data records to read at a time. This parameter determines the number of interactions between Data Integration and the source database and affects read efficiency. **Note** If you set this parameter to a value greater than 2048, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

Configure DM Reader by using the codeless UI

This method is not supported.

Configure DM Reader by using the code editor

In the following code, a synchronization node is configured to read data from a DM database. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
                  "concurrent":1 // The maximum number of parallel threads.
                  "mbps":"12",// The maximum transmission rate.
        }
    },
    "steps": [
        {
            "category": "reader",
            "name": "Reader",
            "parameter": {
                "connection": [
                    {
                        "jdbcUrl": [
                            "jdbc:dm://ip:port/database"
                        ],
                        "table": [
                            "table"
                        ]
                    }
                ],
                "username": "username",
                "password": "password",
                "table": "table",
                "column": [
                    "*"
                ],
                "preSql": [
                    "delete from XXX;"
                ]
            },
            "stepType": "rdbms"
        },
        {
            "category": "writer",
            "name": "Writer",
            "parameter": {},
            "stepType": "stream"
        }
    ],
    "type": "job",
    "version": "2.0"
}
```

5.2.40. Amazon S3 Reader

Amazon Simple Storage Service (Amazon S3) Reader is used to read data from Amazon S3 buckets. This topic describes the data types and parameters that are supported by Amazon S3 Reader and how to configure Amazon S3 Reader by using the codeless user interface (UI) and the code editor.

Background information

Amazon S3 Reader reads data stored in Amazon S3 buckets. Amazon S3 Reader uses Amazon S3 SDK for Java provided by Amazon to read data from Amazon S3. Then, Amazon S3 Reader converts the data to a format that is readable to Data Integration and sends the converted data to a writer.

Amazon S3 stores unstructured data. Amazon S3 Reader provides the following features:

- Reads data from TXT objects. The data in the TXT objects must be logical two-dimensional tables.
- Reads data from CSV-like objects with custom delimiters.
- Reads data of various types as strings and supports constants and column pruning.
- Supports recursive data read and object name-based filtering.
- Supports object compression. The following compression formats are supported: GZIP, BZIP2, and ZIP.
- Uses parallel threads to read data from multiple objects at the same time.

Limits

- Amazon S3 data sources in the Chinese mainland and Hong Kong (China) are not supported.
- Amazon S3 Reader does not support the following features:
  - Uses parallel threads to read data from a single object.
  - Uses parallel threads to read data from a compressed object.
  - Reads data from an object that exceeds 100 GB in size.

Data types

Category	Data Integration data type	Amazon S3 data type
Integer	LONG	LONG
Floating point	DOUBLE	DOUBLE
String	STRING	STRING
Date and time	DATE	DATE
Boolean	BOOL	BOOL

Configure Amazon S3 Reader by using the code editor

- Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	N/A

Parameter	Description	Required	Default value
Object	The name of the Amazon S3 object. You can specify multiple objects from which Amazon S3 Reader reads data. For example, if a bucket contains the *test* folder in which the *ll.txt* object resides, the name of this object is *test/ll.txt*. - If you specify a single Amazon S3 object, Amazon S3 Reader uses a single thread to read data. - If you specify multiple Amazon S3 objects, Amazon S3 Reader uses parallel threads to read data. The number of threads is determined by the number of channels. - If you specify a name that contains a wildcard, Amazon S3 Reader reads data from all objects that match the name. For example, if you set this parameter to abc[0-9], Amazon S3 Reader reads data from objects abc0 to abc9. We recommend that you do not use wildcards because an out of memory (OOM) error may occur. **Note** - Data Integration considers all objects that are read in a synchronization node as a single table. Make sure that all objects that are read in a synchronization node use the same schema. - Control the number of objects stored in a folder. If a folder contains excessive objects, an OOM error may occur. In this case, store the objects in different folders before you synchronize data.	Yes	N/A

column	The columns from which you want to read data. The type parameter specifies the source data type. The index parameter specifies the ID of the column in the source object, starting from 0. The value parameter specifies the column value if the column is a constant column. Amazon S3 Reader does not read a constant column from the source. Instead, Amazon S3 Reader generates a constant column based on the value that you specify.
	You can specify the column parameter in the following format. In this case, Amazon S3 Reader reads all data as strings.
	`column": ["*"]`
	You can also specify a column to read and a constant column in the following format:
	`"column":`
	`{`
	`"type": "long",`
	`"index": 0 // The first INT-type column in the object from which you want to read data.`
	`},`
	`{`
	`"type": "string",`
	`"value": "alibaba" // The value of the current column. In this code, the value is the constant alibaba.`
	`}`
	Note In the column parameter, you must specify the type parameter and specify the index or value parameter.
		Yes	*, which indicates that Amazon S3 Reader reads all data as strings.

Parameter	Description	Required	Default value
fieldDelimiter	The column delimiter that is used in the Amazon S3 object from which you want to read data. **Note** Amazon S3 Reader uses a column delimiter to read data. The default column delimiter is a comma (,). If you do not specify the column delimiter, the default column delimiter is used. If the delimiter is non-printable, enter a value encoded in Unicode, such as \u001b or \u007c.	Yes	Comma (,)
compress	The format in which objects are compressed. By default, this parameter is left empty, which means that objects are not compressed. Amazon S3 Reader supports the following compression formats: GZIP, BZIP2, and ZIP.	No	Empty
encoding	The encoding format of the objects from which you want to read data.	No	utf-8
nullFormat	The string that represents a null pointer. No standard strings can represent a null pointer in TXT objects. You can use this parameter to define a string that represents a null pointer. For example, if you set the `nullFormat` parameter to `null`, Amazon S3 Reader considers null as a null pointer. You can escape empty strings in the following format: `\N=\\N`.	No	N/A
skipHeader	Specifies whether to skip the headers in a CSV-like object. Valid values: ○ True: Amazon S3 Reader reads the headers in a CSV-like object. ○ False: Amazon S3 Reader ignores the headers in a CSV-like object. **Note** The **skipHeader** parameter is unavailable for compressed objects.	No	false
csvReaderConfig	The configurations required to read CSV-like objects. The parameter value must be of the MAP type. A CSV-like object reader is used to read data from CSV-like objects. The CSV-like object reader supports multiple configurations. If no configuration is specified, the default settings are used.	No	N/A

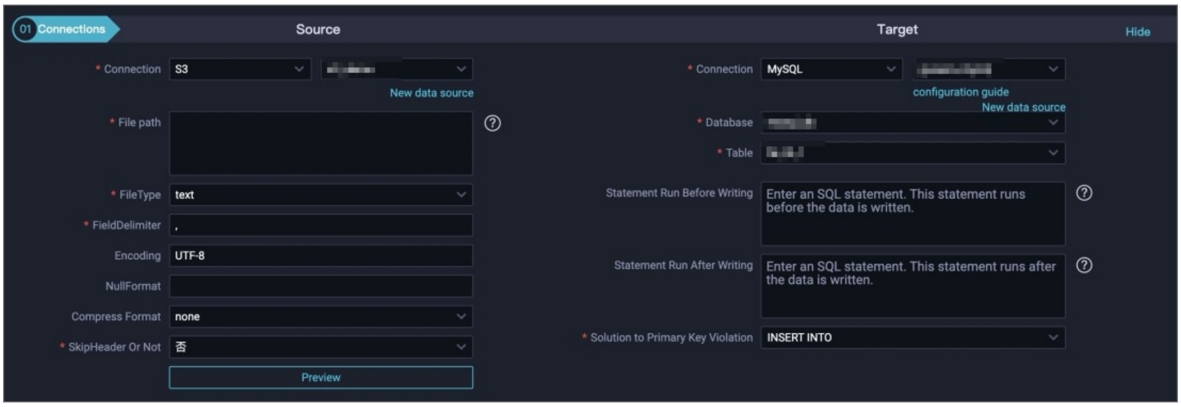
- In the following sample code, a synchronization node is configured to read data from an Amazon S3 bucket. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor. The following code provides a sample script:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"s3",// The reader type.
            "parameter":{
                "nullFormat":"",// The string that represents a null pointer.
                "compress":"",// The format in which objects are compressed.
                "datasource":"",// The name of the data source.
                "column":[// The columns from which you want to read data.
                    {
                        "index":0,// The ID of a column in the source object.
                        "type":"string"// The data type of the column.
                    },
                    {
                        "index":1,
                        "type":"long"
                    },
                    {
                        "index":2,
                        "type":"double"
                    },
                    {
                        "index":3,
                        "type":"boolean"
                    },
                    {
                        "format":"yyyy-MM-dd HH:mm:ss", // The time format.
                        "index":4,
                        "type":"date"
                    }
                ],
                "skipHeader":"",// Specifies whether to skip the headers in a CSV-like object.
                "encoding":"",// The encoding format.
                "fieldDelimiter":",",// The column delimiter.
                "fileFormat": "",// The format of the object.
                "object":[]// The name of the object from which you want to read data.
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":""// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. A value of false indicates that bandwidth throttling is disabled, and a value of true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1 // The maximum number of parallel threads.
            "mbps":"12",// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

Configure Amazon S3 Reader by using the codeless UI

1. Configure data sources.

Set parameters in the **Source** and **Target** section for the synchronization node.



Source

- * Connection: S3 [redacted]
- * File path:
- * FileType: text
- * FieldDelimiter: ,
- Encoding: UTF-8
- NullFormat:
- Compress Format: none
- * SkipHeader Or Not: 否

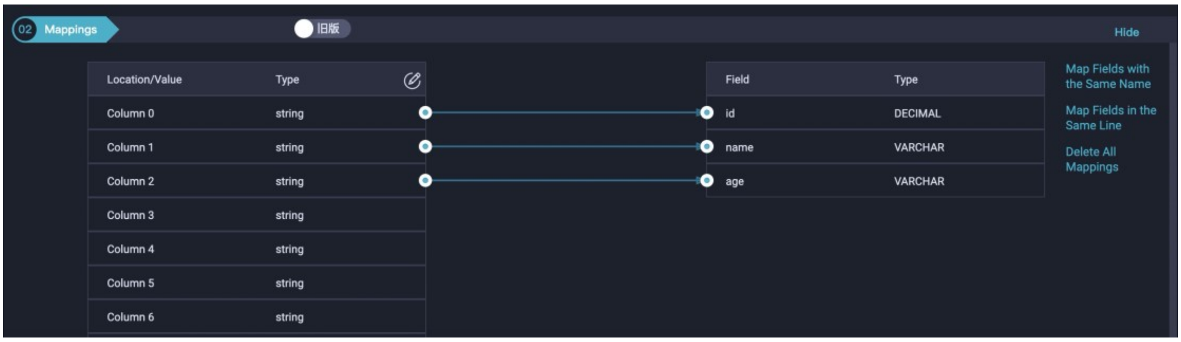
Target

- * Connection: MySQL [redacted]
- * Database: [redacted]
- * Table: [redacted]
- Statement Run Before Writing: Enter an SQL statement. This statement runs before the data is written.
- Statement Run After Writing: Enter an SQL statement. This statement runs after the data is written.
- * Solution to Primary Key Violation: INSERT INTO

Parameter	Description
Connection	The name of the data source from which you want to read data. This parameter is equivalent to the datasource parameter that you set when you use the code editor.
Object Name (Path Included)	The name of the object from which you want to read data. This parameter is equivalent to the Object parameter that you set when you use the code editor. **Note** If an Amazon S3 object is named based on the date, such as *aaa/20171024abc.txt*, you can set this parameter to *aaa/${bdp.system.bizdate}abc.txt*.
Field Delimiter	The column delimiter. This parameter is equivalent to the fieldDelimiter parameter that you set when you use the code editor. By default, a comma (,) is used as the column delimiter.
Encoding	The encoding format. This parameter is equivalent to the encoding parameter that you set when you use the code editor. Default value: UTF-8.
Null String	The string that represents a null pointer. This parameter is equivalent to the nullFormat parameter that you set when you use the code editor. If the source data contains the specified string, the string is replaced with null.
Compression Format	The format in which objects are compressed. This parameter is equivalent to the compress parameter that you set when you use the code editor. By default, objects are not compressed.
Include Header	Specifies whether to skip the headers in the object. This parameter is equivalent to the skipHeader parameter that you set when you use the code editor. Default value: No.

2. Configure field mappings. This operation is equivalent to setting the column parameter when you use the code editor.

   Fields in the source on the left have a one-to-one mapping with fields in the destination on the right. You can click **Add** to add a field. To remove an added field, move the pointer over the field and click the **Remove** icon.

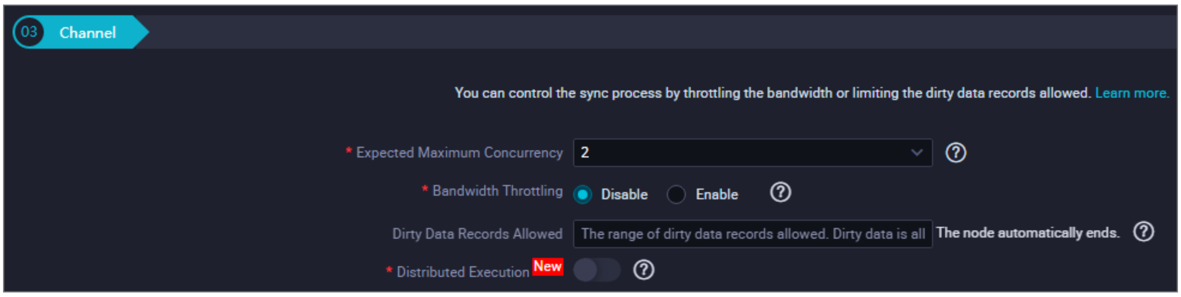


Location/Value	Type
Column 0	string
Column 1	string
Column 2	string
Column 3	string
Column 4	string
Column 5	string
Column 6	string

Field	Type
id	DECIMAL
name	VARCHAR
age	VARCHAR

Parameter	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that have been established.

3. Configure channel control policies.



03 Channel

You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node can use to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	Not supported.

5.3. Configure a writer

5.3.1. Configure AnalyticDB for MySQL 2.0 Writer

This topic describes the data types and parameters supported by AnalyticDB for MySQL 2.0 Writer and how to configure it by using the codeless user interface (UI) and code editor.

Data Integration can import data to AnalyticDB for MySQL 2.0 in real time. This method requires you to create real-time tables, which are fact tables, in the destination AnalyticDB for MySQL 2.0 database in advance. In real-time import mode, data is imported efficiently and the process is simple.

You must configure a connection before configuring AnalyticDB for MySQL 2.0 Writer. For more information, see Add an AnalyticDB for MySQL 2.0 data source.

The following table lists the data types supported by AnalyticDB for MySQL 2.0 Writer.

Type	AnalyticDB for MySQL 2.0 data type
Integer	INT, TINYINT, SMALLINT, and BIGINT
Floating point	FLOAT and DOUBLE
String	VARCHAR
Date and time	DATE and TIMESTAMP

Type	AnalyticDB for MySQL 2.0 data type
Boolean	BOOLEAN

Parameters

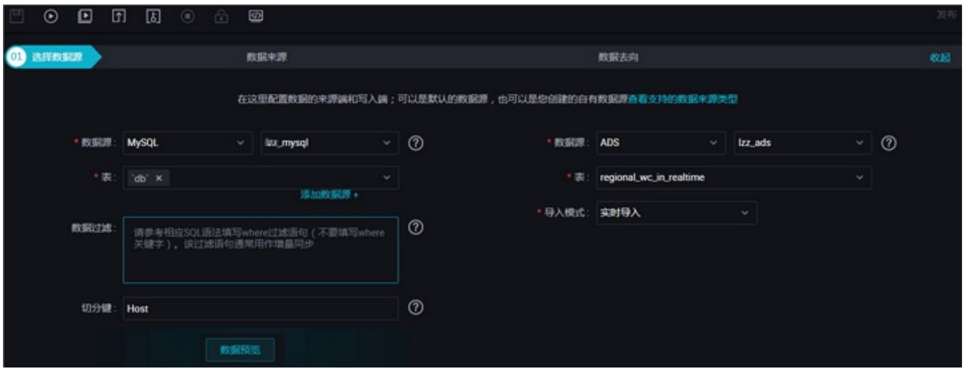
Parameter	Description	Required	Default value
url	The URL for connecting to the AnalyticDB for MySQL 2.0 database. Specify the parameter in the Address:Port format.	Yes	None
database	The name of the AnalyticDB for MySQL 2.0 database.	Yes	None
Access Id	The AccessKey ID used for connecting to the AnalyticDB for MySQL 2.0 database.	Yes	None
Access Key	The AccessKey secret used for connecting to the AnalyticDB for MySQL 2.0 database.	Yes	None
datasource	The connection name. It must be identical to the name of the added connection. You can add connections in the code editor.	Yes	None
table	The name of the destination table.	Yes	None
partition	The partition name of the destination table. If the destination table is partitioned, this parameter is required.	No	None
writeMode	The write mode. Set the value to insert. In this mode, if a primary key conflict occurs, the conflicting rows are overwritten.	Yes	None
column	The columns in the destination table to which data is written. Separate the columns with commas (,), for example, ["a", "b", "c"]. Set the value to an asterisk (*) if data is written to all the columns in the destination table.	Yes	None
suffix	Optional. The suffix to the AnalyticDB for MySQL 2.0 URL that is in the format of `Address:Port`. This suffix is a custom connection string. For more information, see the Java Database Connectivity (JDBC) URL parameters supported by MySQL. After this parameter is set, the URL changes to a JDBC connection string for accessing AnalyticDB for MySQL 2.0. For example, set the suffix parameter to `autoReconnect=true&failOverReadOnly=false&maxReconnects=10`.	No	None
batchSize	The number of data records to write at a time.	The parameter is required and takes effect only when the writeMode parameter is set to insert.	None

Parameter	Description	Required	Default value
bufferSize	The size of the Data Integration data buffer, which is designed to improve the performance of AnalyticDB for MySQL 2.0. Data from the source database is sorted in the buffer before being committed to AnalyticDB for MySQL 2.0. The data in the buffer is sorted based on the partition key columns in AnalyticDB for MySQL 2.0. In this way, the data is organized in an order that is friendly to the AnalyticDB for MySQL 2.0 server. The data in the buffer is committed to AnalyticDB for MySQL 2.0 in batches based on the batchSize parameter. We recommend that you set the bufferSize parameter to a multiple of the value of the batchSize parameter.	The parameter is required and takes effect when the writeMode parameter is set to insert.	Disabled by default

Configure AnalyticDB for MySQL 2.0 Writer by using the codeless UI

1. Configure the connections.

   Configure the source and destination connections for the sync node.



在这里配置数据的来源端和写入端；可以是默认的数据源，也可以是您创建的自有数据源查看支持的数据来源类型

数据来源：

* 数据源： MySQL lzz_mysql
* 表： `db`
数据过滤： 请参考相应SQL语法填写where过滤语句（不要填写where关键字），该过滤语句通常用作增量同步
切分键： Host

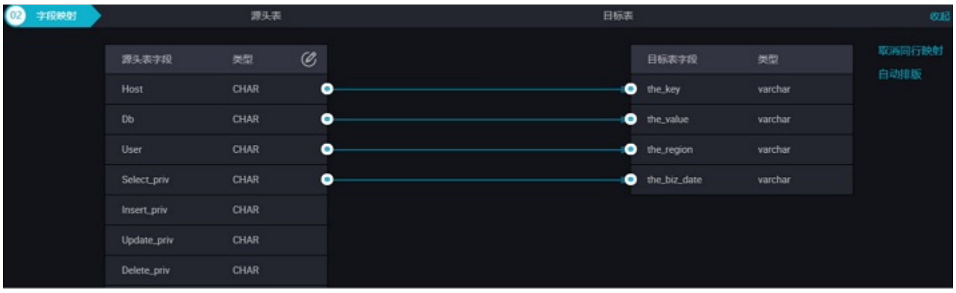
数据去向：

* 数据源： ADS lzz_ads
* 表： regional_wc_in_realtime
* 导入模式： 实时导入

Parameter	Description
Connection	The connection name. In this example, select AnalyticDB for MySQL 2.0.
Table	The name of the destination table. Select a table in the AnalyticDB for MySQL 2.0 database to which data is synchronized from the source database.
Write Method	The method in which data is written to the destination table. Select the write method based on the update mode of the destination table in AnalyticDB for MySQL 2.0. Valid values: **Batch** and **Real-Time**. **Note** The Batch mode is supported only for synchronizing data from MaxCompute to AnalyticDB for MySQL 2.0. To import data in batches, configure two sync nodes. Configure one sync node to write data in batches to MaxCompute. Configure the other sync node to synchronize data in batches from MaxCompute to AnalyticDB for MySQL 2.0.
Writing Rule	The writing rule. If you select **Write with Original Data Deleted**, all data in the table or partition is cleared before new data is imported. This rule is equivalent to the `INSERT OVERWRITE` statement.
Partition Key Column 1	The partition to which data is written. The default value cannot be modified.

2. Configure field mapping, that is, the column parameter in the preceding parameter description.

Fields in the source table on the left have a one-to-one mapping with fields in the destination table on the right. You can click **Add** to add a field, or move the pointer over a field and click the **Delete** icon to delete the field.

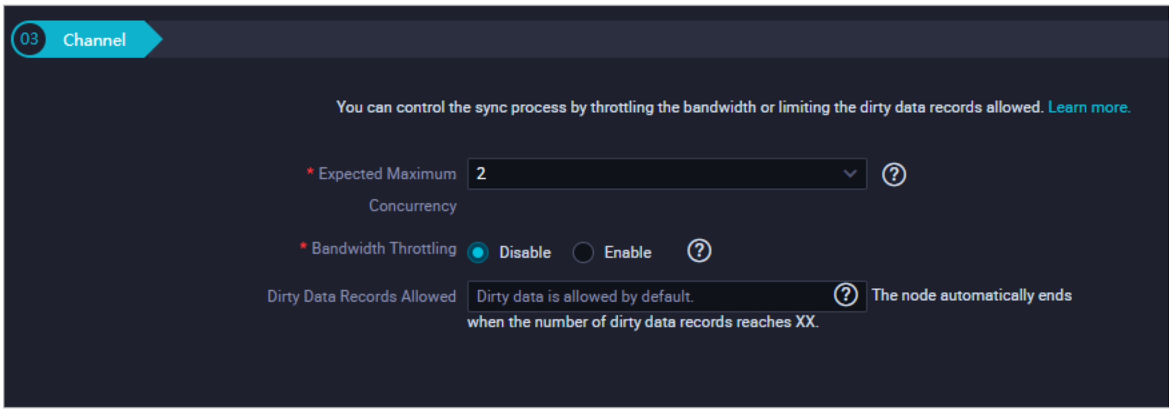


源头表字段	类型
Host	CHAR
Db	CHAR
User	CHAR
Select_priv	CHAR
Insert_priv	CHAR
Update_priv	CHAR
Delete_priv	CHAR

目标表字段	类型
the_key	varchar
the_value	varchar
the_region	varchar
the_biz_date	varchar

Button or icon	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish a mapping between fields with the same name. Note that the data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish a mapping for fields in the same row. Note that the data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove mappings that have been established.
Auto Layout	Click Auto Layout. The fields are automatically sorted based on specified rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box that appears, you can manually edit fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	○ Click Add to add a field. You can enter constants. Each constant must be enclosed in single quotation marks (' '), such as 'abc' and '123'. ○ You can use scheduling parameters, such as ${bizdate}. ○ You can enter functions supported by relational databases, such as now() and count(1). ○ Fields that cannot be parsed are indicated by Unidentified.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: Dirty data is allowed by default. The node automatically ends when the number of dirty data records reaches XX.

Parameter	Description
Expected Maximum Concurrency	The maximum number of concurrent threads to read data from or write data to data storage within the sync node. You can configure the concurrency for a node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.

Parameter	Description
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Resource Group	The resource group used for running the sync node. If a large number of nodes including this sync node are deployed on the default resource group, the sync node may need to wait for resources. We recommend that you purchase an exclusive resource group for data integration or add a custom resource group.

Configure AnalyticDB for MySQL 2.0 Writer by using the code editor

```
{
    "type":"job",
    "version":"2.0",
    "steps":[// The following template is used to configure Stream Reader. For more information, see the corres
ponding topic.
        {
            "stepType":"stream",
            "parameter":{
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"ads",// The writer type.
            "parameter":{
                "partition":"",// The name of the destination partition in the destination table.
                "datasource":"",// The connection name.
                "column":[// The columns to which data is written.
                     "id"
                ],
                "writeMode":"insert",// The write mode.
                "batchSize":"256",// The number of data records to write at a time.
                "table":"",// The name of the destination table.
                "overWrite":"true"// Specifies whether to overwrite the destination table when data is written
to AnalyticDB for MySQL 2.0. A value of true indicates that the destination table is overwritten. A value of fa
lse indicates that the destination table is not overwritten and the new data is appended to the existing data.
This value only takes effect when the writeMode parameter is set to load.
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":false,// Specifies whether to enable bandwidth throttling. A value of false indicates th
at the bandwidth is not throttled. A value of true indicates that the bandwidth is throttled. The maximum trans
mission rate takes effect only if you set this parameter to true.
            "concurrent":1,// The maximum number of concurrent threads.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.2. DataHub Writer

This topic describes the parameters that are supported by DataHub Writer and how to configure DataHub Writer by using the codeless user interface (UI) and code editor.

DataHub is a real-time data distribution platform that is designed to process streaming data. You can publish and subscribe to streaming data in DataHub and distribute the data to other platforms. This allows you to analyze streaming data and build applications based on the streaming data.

DataHub is built on top of the Apsara distributed operating system, and features high availability, low latency, high scalability, and high throughput. DataHub is seamlessly integrated with Realtime Compute for Apache Flink, and allows you to use SQL statements to analyze streaming data. DataHub can also distribute streaming data to Alibaba Cloud services, such as MaxCompute and Object Storage Service (OSS).

> **Notice** Strings must be encoded in the UTF-8 format. The size of each string must not exceed 1 MB.

Channel types

The source is connected to the sink by using a single channel. Therefore, the channel type configured for the writer must be the same as that configured for the reader. In normal cases, channels are categorized into two types: memory and file. In the following configuration, the channel type is set to file:

```
"agent.sinks.dataXSinkWrapper.channel": "file"
```

Parameters

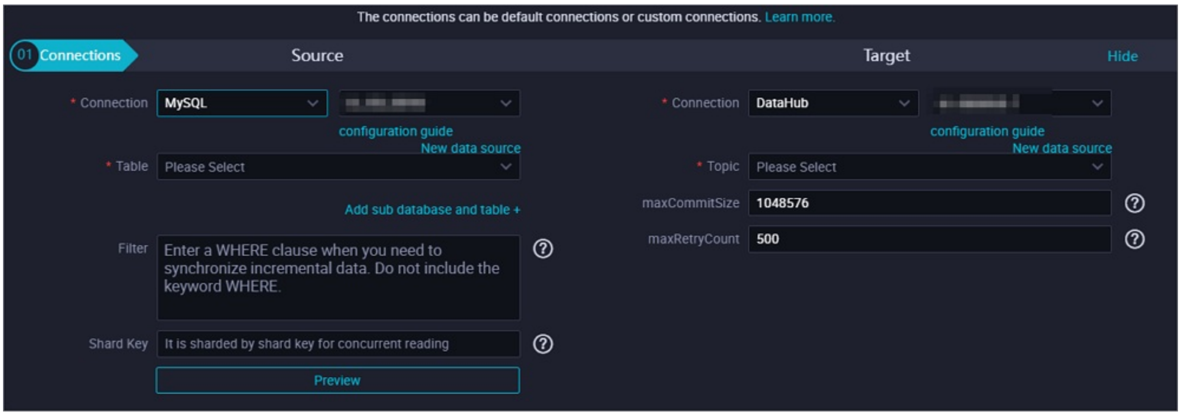
Parameter	Description	Required	Default value
accessId	The AccessKey ID of the account that you use to connect to DataHub.	Yes	No default value
accessKey	The AccessKey secret of the account that you use to connect to DataHub.	Yes	No default value
endPoint	The endpoint of DataHub.	Yes	No default value
maxRetryCount	The maximum number of retries if the synchronization node fails.	No	No default value
mode	The mode for writing strings.	Yes	No default value
parseContent	The data to be parsed.	Yes	No default value
project	The basic organizational unit of data in DataHub. Each project has one or more topics. **Note** DataHub projects are independent of MaxCompute projects. You cannot use MaxCompute projects as DataHub projects.	Yes	No default value
topic	The minimum unit for data subscription and publishing. You can use topics to distinguish different types of streaming data.	Yes	No default value

Parameter	Description	Required	Default value
maxCommitSize	The maximum amount of the buffered data that Data Integration can accumulate before it commits the data to the destination. You can specify this parameter to improve writing efficiency. The default value is 1048576, in bytes, which is 1 MB. DataHub allows for a maximum of 10,000 data records to be written in a single write request. If the number of data records exceeds 10,000, the synchronization node fails. You can control the number of data records to be written in a single write request based on the total amount of data that is calculated by using the following formula: Average amount of data in a single data record × 10,000.	No	*1MB*

Configure DataHub Writer by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



The connections can be default connections or custom connections. Learn more.

01 Connections

Source

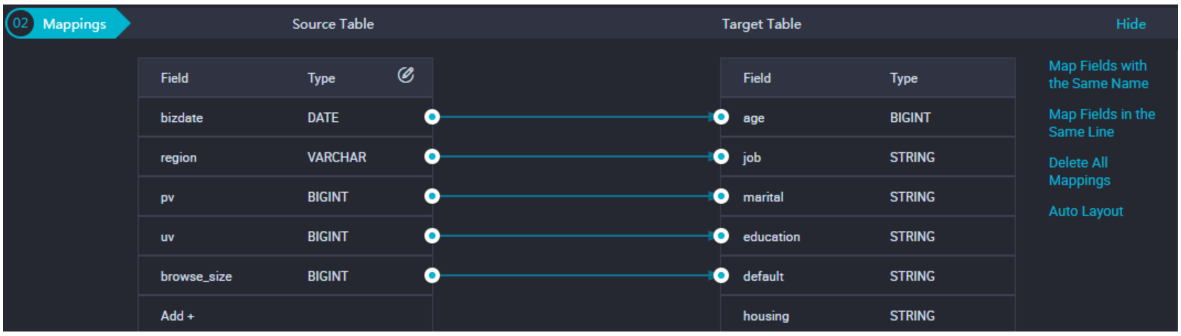
- * Connection: MySQL [redacted]
- * Table: Please Select
- Filter: Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.
- Shard Key: It is sharded by shard key for concurrent reading

Target

- * Connection: DataHub [redacted]
- * Topic: Please Select
- maxCommitSize: 1048576
- maxRetryCount: 500

Parameter	Description
Connection	The name of the data source to which you want to write data.
Topic	This parameter is equivalent to the topic parameter that is described in the preceding section.
maxCommitSize	The maximum amount of the data that is written to DataHub in a single request. Unit: bytes.
maxRetryCount	This parameter is equivalent to the maxRetryCount parameter that is described in the preceding section.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.



02 Mappings

Source Table Field	Type	Target Table Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
Add +		housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3.

Configure DataHub Writer by using the code editor

In the following code, a synchronization node is configured to write data from memory to DataHub by using the code editor. For more information, see Create a synchronization node by using the code editor.

```
{
    "type": "job",
    "version": "2.0",// The version number.
    "steps": [
        {
            "stepType": "stream",
            "parameter": {},
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "datahub",// The writer type.
            "parameter": {
                "datasource": "",// The name of the data source to which you want to write data.
                "topic": "",// The minimum unit for data subscription and publishing. You can use topics to dis
tinguish different types of streaming data.
                "maxRetryCount": 500,// The maximum number of retries if the synchronization node fails.
                "maxCommitSize": 1048576// The maximum amount of the buffered data that Data Integration can ac
cumulate before it commits the data to the destination.
                 // DataHub allows for a maximum of 10,000 data records to be written in a single write request
. If the number of data records exceeds 10,000, the synchronization node fails. You can control the number of d
ata records to be written in a single write request based on the total amount of data that is calculated by usi
ng the following formula: Average amount of data in a single data record × 10,000. For example, if the data siz
e of a single data record is 10 KB, the value of this parameter must be less than the result of 10 multiplied b
y 10,000.
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "setting": {
        "errorLimit": {
            "record": ""// The maximum number of dirty data records allowed.
        },
        "speed": {
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":20, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}
```

5.3.3. DB2 Writer

This topic describes the data types and parameters that are supported by Db2 Writer and how to configure Db2 Writer by using the codeless user interface (UI) and code editor.

Notice Db2 Writer supports only exclusive resource groups for Data Integration, but not shared resource groups or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration, Use the default resource group, and Add a custom resource group for Data Integration.

Background information

Db2 Writer writes data to tables stored in Db2 databases. Db2 Writer connects to a remote Db2 database by using Java Database Connectivity (JDBC), and executes an `INSERT INTO` statement to write data to the Db2 database. The data is written to the Db2 database in batches.

Db2 Writer is designed for extract, transform, load (ETL) developers to import data from data warehouses to Db2 databases. Db2 Writer can also be used as a data migration tool by users such as database administrators.

Db2 Writer obtains data from a reader and writes the data to the destination database by executing the `INSERT INTO` statement. If a primary key conflict or unique index conflict occurs, data cannot be written to the conflicting rows. To improve performance, Db2 Writer executes `batch updates by using a PreparedStatement object` and sets the `rewriteBatchedStatements` parameter to true. This way, Data Integration commits a write request when the amount of the buffered data that it accumulates reaches a specific threshold.

Note A synchronization node that uses Db2 Writer must have at least the permissions to execute the `INSERT INTO` statement. Whether other permissions are required depends on the SQL statements that you specify in the preSql and postSql parameters when you configure the node.

The version of the Db2 JDBC driver that Db2 Writer uses is IBM Data Server Driver for JDBC and SQLJ 4.11.77. For more information about the mappings between the versions of Db2 JDBC drivers and the Db2 versions, see IBM Support.

Data types

Db2 Writer supports most Db2 data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by Db2 Writer.

Category	Db2 data type
Integer	SMALLINT
Floating point	DECIMAL, REAL, and DOUBLE
String	CHAR, CHARACTER, VARCHAR, GRAPHIC, VARGRAPHIC, LONG VARCHAR, CLOB, LONG VARGRAPHIC, and DBCLOB
Date and time	DATE, TIME, and TIMESTAMP
Boolean	N/A
Binary	BLOB

Parameters

Parameter	Description	Required	Default value
jdbcUrl	The JDBC URL of the Db2 database. In accordance with the official Db2 specifications, the URL must be in the jdbc:db2://ip:port/database format. You can also specify the information of the attachment facility.	Yes	No default value
username	The username that you use to connect to the Db2 database.	Yes	No default value
password	The password that you use to connect to the Db2 database.	Yes	No default value
table	The name of the table to which you want to write data.	Yes	No default value
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as "column": ["id", "name", "age"]. If you want to write data to all the columns in the destination table, set this parameter to an asterisk (*), such as "column": ["*"].	Yes	No default value

Parameter	Description	Required	Default value
preSql	The SQL statement that you want to execute before the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to delete outdated data. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to add a timestamp. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and Db2 and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1024*

Configure Db2 Writer by using the codeless UI

This method is not supported.

Configure Db2 Writer by using the code editor

In the following code, a synchronization node is configured to write data to a Db2 database by using the code editor. For more information, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"db2",// The writer type.
            "parameter":{
                "postSql":[],// The SQL statement that you want to execute after the synchronization node is ru
n.
                "password":"",// The password that you use to connect to the Db2 database.
                "jdbcUrl":"jdbc:db2://ip:port/database",// The JDBC URL of the Db2 database.
                "column":[
                    "id"
                ],
                "batchSize":1024,// The number of data records to write at a time.
                "table":"",// The name of the table to which you want to write data.
                "username":"",// The username that you use to connect to the Db2 database.
                "preSql":[]// The SQL statement that you want to execute before the synchronization node is run
.
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.4. DRDS Writer

This topic describes the data types and parameters that are supported by DRDS Writer and how to configure DRDS Writer by using the codeless user interface (UI) and code editor.

Background information

DRDS Writer writes data to tables that are stored in Distributed Relational Database Service (DRDS) databases. DRDS Writer connects to the proxy of a remote DRDS database by using Java Database Connectivity (JDBC) and executes the `REPLACE INTO` statement to write data to the DRDS database.

Note

- To execute the `REPLACE INTO` statement, you must make sure that your table has the primary key or a unique index to prevent duplicate data.
- Before you configure DRDS Writer, you must add a DRDS data source. For more information, see Add a DRDS data source.
- DataWorks does not support DRDS instances that run MySQL 8.0.

DRDS Writer is designed for extract, transform, load (ETL) developers to import data in data warehouses to DRDS databases. DRDS Writer can also be used as a data migration tool by users such as database administrators.

DRDS Writer obtains data from a reader and executes the `REPLACE INTO` statement to write the data to the destination database. If no primary key conflict or unique index conflict occurs, data is directly written to the destination table, which is the same as the action of the `INSERT INTO` statement. If a conflict occurs, data in conflicting rows in the destination table is replaced by new data. DRDS Writer sends data to the DRDS proxy when the amount of buffered data reaches a specific threshold. The proxy determines whether to write the data to one or more tables and how to route the data when the data is written to multiple tables.

Note A synchronization node that uses DRDS Writer must have at least the permissions to execute the `REPLACE INTO` statement. Whether other permissions are required depends on the SQL statements that you specify in the preSql and postSql parameters when you configure the node.

Data types

DRDS Writer supports most DRDS data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by DRDS Writer.

Category	DRDS data type
Integer	INT, TINYINT, SMALLINT, MEDIUMINT, BIGINT, and YEAR
Floating point	FLOAT, DOUBLE, and DECIMAL
String	VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT
Date and time	DATE, DATETIME, TIMESTAMP, and TIME
Boolean	BIT and BOOLEAN
Binary	TINYBLOB, MEDIUMBLOB, BLOB, LONGBLOB, and VARBINARY

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table to which you want to write data.	Yes	No default value
writeMode	The write mode. Valid values: • *insert ignore*: If a primary key conflict or unique index conflict occurs, the source data cannot be written. • *replace into*: If a primary key conflict or unique index conflict occurs, the original data is deleted, and new data is inserted.	No	*insert ignore*

Parameter	Description	Required	Default value
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as "column": ["id","name","age"]. If you want to write data to all the columns in the destination table, set this parameter to an asterisk (*), such as "column": ["*"].	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor. For example, you can set this parameter to `delete * from table xxx;` to delete data from the xxx table before data synchronization. You can specify this parameter based on your business requirements.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor. For example, you can set this parameter to `delete * from table xxx where xx=xx;` to delete specific data from the xxx table after data synchronization. You can specify this parameter based on your business requirements.	No	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and DRDS and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

Configure DRDS Writer by using the codeless UI

Create a synchronization node and configure the node. For more information, see Configure a synchronization node by using the codeless UI.

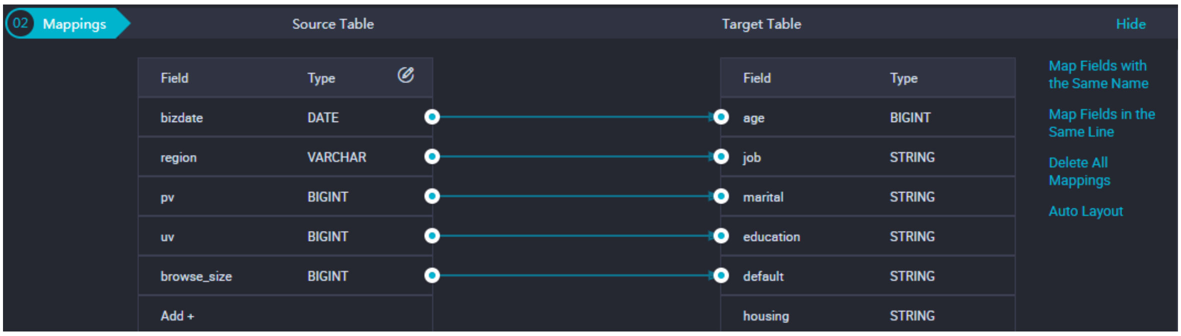
Perform the following steps on the configuration tab of the node:

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table to which you want to write data. This parameter is equivalent to the table parameter that is described in the preceding section.
Statement Run Before Writing	The SQL statement that you want to execute before the synchronization node is run. This parameter is equivalent to the preSql parameter that is described in the preceding section.
Statement Run After Writing	The SQL statement that you want to execute after the synchronization node is run. This parameter is equivalent to the postSql parameter that is described in the preceding section.
Solution to Primary Key Violation	The write mode. This parameter is equivalent to the writeMode parameter that is described in the preceding section. You can select the desired write mode.

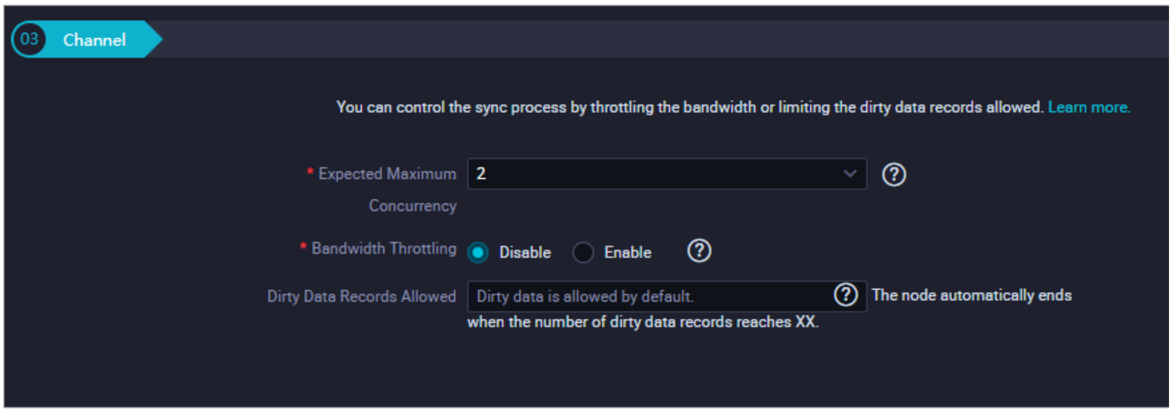
2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.



Field	Type	Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
		housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: Dirty data is allowed by default. The node automatically ends when the number of dirty data records reaches XX.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.

4.

Configure DRDS Writer by using the code editor

In the following code, a synchronization node is configured to write data to a DRDS database. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
                },
        {
            "stepType":"drds",// The writer type.
            "parameter":{
                "postSql":[],// The SQL statement that you want to execute after the synchronization node is ru
n.
                "datasource":"",// The name of the data source.
                "column":[// The names of the columns to which you want to write data.
                "id"
                ],
                "writeMode":"insert ignore",
                "batchSize":"1024",// The number of data records to write at a time.
                "table":"test",// The name of the table to which you want to write data.
                "preSql":[]// The SQL statement that you want to execute before the synchronization node is run
.
                },
            "name":"Writer",
            "category":"writer"
                }
                ],
    "setting":{
        "errorLimit":{
        "record":"0"// The maximum number of dirty data records allowed.
            },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
                }
            },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
                }
            ]
        }
    }
```

5.3.5. FTP Writer

This topic describes the parameters that are supported by FTP Writer and how to configure FTP Writer by using the codeless user interface (UI) and code editor.

FTP Writer writes one or more comma-separated values (CSV) files to a remote File Transfer Protocol (FTP) server. FTP Writer converts the data that is obtained from a reader to CSV files. Then, FTP Writer writes these files to a remote FTP server by using FTP-related network protocols.

Note Before you configure FTP Writer, you must add an FTP data source. For more information, see Add an FTP data source.

FTP Writer can write files that store logical two-dimensional tables, such as CSV files that store text data, to an FTP server.

FTP Writer converts the data that is obtained from a reader to files and writes the files to an FTP server. The files on the FTP server store only unstructured data. FTP Writer provides the following features:

- Writes only text files to an FTP server. The data in the files must be organized as logical two-dimensional tables. FTP Writer cannot write files that store binary large object (BLOB) data, such as video data, to an FTP server.
- Writes TXT and CSV-like files that contain custom delimiters to an FTP server.
- Writes uncompressed files to an FTP server.
- Uses parallel threads to write files to an FTP server. Each thread writes a file.

FTP Writer does not support the following features:

- Uses parallel threads to write a single file to an FTP server.
- Distinguishes between data types. FTP does not distinguish between data types. Therefore, FTP Writer writes all data as strings to files on an FTP server.

Parameters

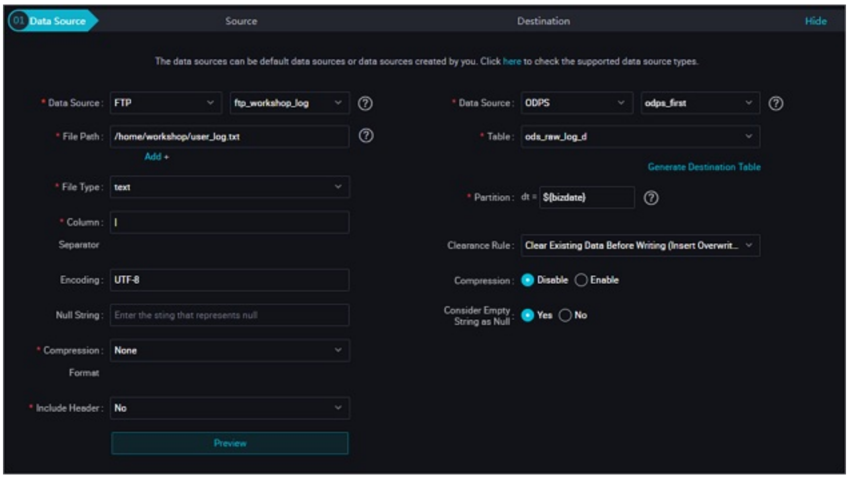
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
timeout	The timeout period of the connection to an FTP server. Unit: milliseconds.	No	*60,000*
path	The directory on the FTP server to which you want to write data. FTP Writer uses parallel threads to write multiple files to the directory based on the parallelism setting.	Yes	No default value
fileName	The name prefix of the files that you want to write to the FTP server. A random suffix is appended to the specified prefix to form the actual file name that is used by each thread.	Yes	No default value
singleFileOutput	Specifies whether to add a random suffix to the names of the files that you want to write to the FTP server. The names of the files that FTP Writer writes to the FTP server are related to the value of the fileName parameter. If you do not need the random suffix, set the singleFileOutput parameter to true. In this case, FTP Writer the files to the FTP server by using the name prefix of the files.	No	*false*
writeMode	The mode in which FTP Writer writes files. Valid values: • *truncate*: FTP Writer deletes all existing files whose names contain the prefix specified by fileName in the destination directory before it writes files to the directory. • *append*: FTP Writer directly writes files based on the file name prefix specified by fileName and ensures that the actual file names do not conflict with the names of existing files. • *nonConflict*: FTP Writer returns an error if the destination directory contains a file whose name contains the prefix specified by fileName.	Yes	No default value
fieldDelimiter	The column delimiter that is used in the files that you want to write to the FTP server. The delimiter must be a single character.	Yes	No default value
skipHeader	Specifies whether to skip the headers in CSV-like files if the files contain headers. The skipHeader parameter is not supported for compressed files.	No	*false*
compress	The compression format of the files that you want to write to the FTP server. Valid values: *gzip* and *bzip2*.	No	Not default value
encoding	The encoding format of the files that you want to write to the FTP server.	No	*utf-8*

Parameter	Description	Required	Default value
nullFormat	The string that represents a null pointer. No standard strings can represent a null pointer in TXT files. You can use this parameter to define which string represents a null pointer. For example, if you set `nullFormat` to null, Data Integration considers null as a null pointer.	No	No default value
dateFormat	The format in which the data of the DATE type is serialized in a file, such as "dateFormat":"yyyy-MM-dd".	No	No default value
fileFormat	The format in which files are written to the FTP server. Valid values: CSV and TEXT. If a file is written as a CSV file, the file must follow CSV specifications. If the data in the file contains column delimiters, the column delimiters are escaped by double quotation marks ("). If a file is written as a TXT file, the data in the file is separated by column delimiters. In this case, the column delimiters are not escaped.	No	*TEXT*
header	The table headers if files are written as TXT or CSV files, such as ["id","name","age"]. This indicates that the id, name, and age fields are written to a CSV file as the first row.	No	No default value
markDoneFileName	The name of the file that is used to indicate that the synchronization node is successfully run. Data Integration checks whether the file exists after data synchronization. Set this parameter to the absolute path of the file.	No	No default value

Configure FTP Writer by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



The data sources can be default data sources or data sources created by you. Click here to check the supported data source types.

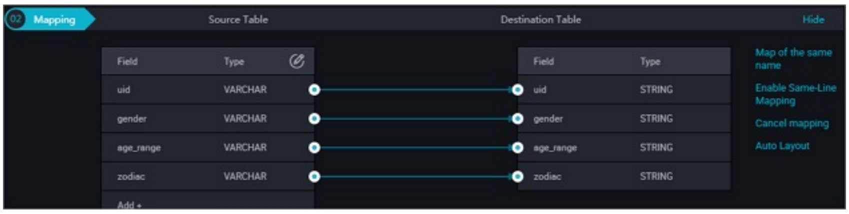
Source:
- * Data Source: FTP, ftp_workshop_log
- * File Path: /home/workshop/user_log.txt
- * File Type: text
- * Column Separator: |
- Encoding: UTF-8
- Null String: Enter the sting that represents null
- * Compression Format: None
- * Include Header: No

Destination:
- * Data Source: ODPS, odps_first
- * Table: ods_raw_log_d
- * Partition: dt = ${bizdate}
- Clearance Rule: Clear Existing Data Before Writing (Insert Overwrit...
- Compression: ● Disable ○ Enable
- Consider Empty String as Null: ● Yes ○ No

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
File Path	This parameter is equivalent to the path parameter that is described in the preceding section.
File Type	The format of the files that you want to write to the FTP server. The default format is csv.
Field Delimiter	This parameter is equivalent to the fieldDelimiter parameter that is described in the preceding section. By default, a comma (,) is used as a column delimiter.

Parameter	Description
Encoding	This parameter is equivalent to the encoding parameter that is described in the preceding section. Default value: *UTF-8*.
Null String	This parameter is equivalent to the nullFormat parameter that is described in the preceding section, which defines a string that represents a null pointer.
Time Format	This parameter is equivalent to the dateFormat parameter that is described in the preceding section.
Solution to Duplicate Prefixes	This parameter is equivalent to the writeMode parameter that is described in the preceding section.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.



Source Table

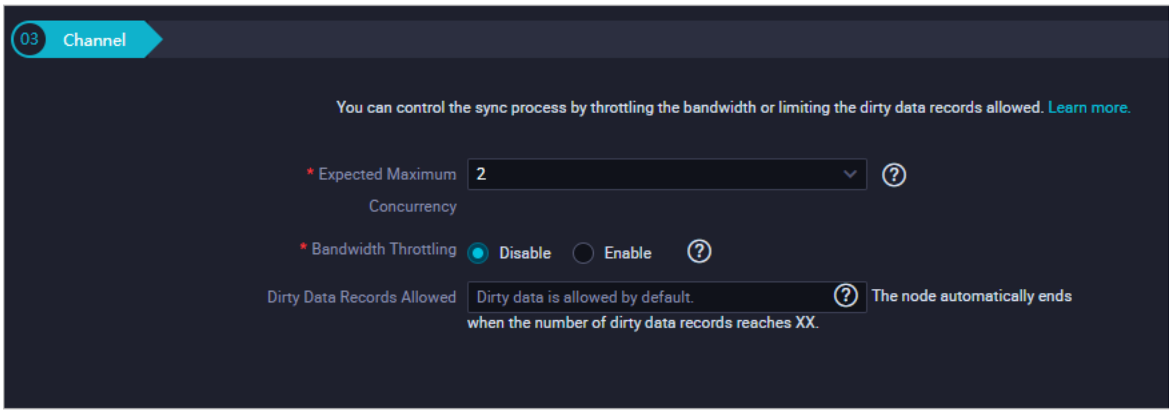
Field	Type
uid	VARCHAR
gender	VARCHAR
age_range	VARCHAR
zodiac	VARCHAR

Destination Table

Field	Type
uid	STRING
gender	STRING
age_range	STRING
zodiac	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: Dirty data is allowed by default. The node automatically ends when the number of dirty data records reaches XX.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.

Parameter	Description
Dirty Data Records Allowed	The maximum number of dirty data records that are allowed.

4.

Configure FTP Writer by using the code editor

In the following code, a synchronization node is configured to write files to an FTP server. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"ftp",// The writer type.
            "parameter":{
                "path":"",// The directory on the FTP server to which you want to write files.
                "fileName":"",// The name prefix of the files that you want to write to the FTP server.
                "nullFormat":"null",// The string that represents a null pointer.
                "dateFormat":"yyyy-MM-dd HH:mm:ss",// The time format.
                "datasource":"",// The name of the data source.
                "writeMode":"",// The write mode.
                "fieldDelimiter":",",// The column delimiter.
                "encoding":"",// The encoding format.
                "fileFormat":""// The format in which FTP Writer writes files.
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.6. HBase Writer

This topic describes the data types and parameters that are supported by HBase Writer and how to configure HBase Writer by using the codeless user interface (UI) and code editor.

HBase Writer writes data to HBase databases. HBase Writer connects to a remote HBase database by using a Java client of HBase and uses the PUT method to write data to the HBase database.

Supported features

- HBase Writer can write data to HBase 0.94.X, HBase 1.1.X, and HBase 2.X.
  - If you use HBase 0.94.X, set the hbaseVersion parameter to 094x.

    ```
    "writer": {
            "hbaseVersion": "094x"
        }
    ```

  - If you use HBase 1.1.X or HBase 2.X, set the hbaseVersion parameter to 11x.

    ```
    "writer": {
            "hbaseVersion": "11x"
        }
    ```

    > **Note** HBase 1.1.X Writer is compatible with HBase 2.0. If you have questions when you use HBase Writer, submit a ticket.

- You can use concatenated fields as a rowkey.

  HBase Writer can concatenate multiple fields to generate the rowkey of an HBase table.

- You can specify the version of each HBase cell.

  Information that can be used as the version of an HBase cell:

  - Current time
  - Specific source column
  - Specific time

Data types

The following table lists the data types that are supported by HBase Writer.

> **Note**
> - The data types of specified columns must be the same as those in an HBase table.
> - Data types that are not listed in the following table are not supported.

Category	HBase data type
Integer	INT, LONG, and SHORT
Floating point	FLOAT and DOUBLE
Boolean	BOOLEAN
String	STRING

Parameters

Parameter	Description	Required	Default value

Parameter	Description	Required	Default value
haveKerberos	Specifies whether Kerberos authentication is required. Valid values: true and false. ⓘ **Note** - If you set this parameter to true, Kerberos authentication is required, and you must configure the following parameters that are related to Kerberos authentication: ○ kerberosKeytabFilePath ○ kerberosPrincipal ○ hbaseMasterKerberosPrincipal ○ hbaseRegionserverKerberosPrincipal ○ hbaseRpcProtection - If you set this parameter to false, Kerberos authentication is not required, and you do not need to configure the preceding parameters.	No	*false*
hbaseConfig	The properties of the HBase cluster, in the JSON format. The hbase.zookeeper.quorum parameter is required. It specifies the ZooKeeper address of the HBase cluster. You can also configure other properties, such as those related to the cache and batch for scan operations. ⓘ **Note** You must use an internal endpoint to access an ApsaraDB for HBase database.	Yes	No default value
mode	The write mode. Only the *normal* mode is supported. The dynamic column mode will be available in the future.	Yes	No default value
table	The name of the HBase table to which you want to write data. The name is case-sensitive.	Yes	No default value
encoding	The encoding format that is used to convert a string to the HBase byte[] format. Valid values: utf-8 and gbk.	No	*utf-8*
column	The names of the columns to which you want to write data. - index: the ID of a column in the source table, starting from 0. - name: the name of a column in the HBase table. Specify this parameter in the format of Column family:Column name. - type: the data type. The value of this parameter is used by the HBase byte[] constructor.	Yes	No default value

<table>
<tr><th>Parameter</th><th>Description</th><th>Required</th><th>Default value</th></tr>
<tr><td>rowkeyColumn</td><td>

The rowkey column of each row in the HBase table.

- index: the ID of a column in the source table, starting from 0. If the column is a constant, set this parameter to -1.
- type: the data type. The value of this parameter is used by the HBase byte[] constructor.
- value: a constant, which is usually used as the delimiter between fields. HBase Writer concatenates all columns that are specified in this parameter to a string in the same order that the columns are specified. Then, HBase Writer uses the string as the rowkey. The specified columns cannot be all constants.

The following code provides a configuration example:

```
"rowkeyColumn": [
          {
            "index":0,
            "type":"string"
          },
          {
            "index":-1,
            "type":"string",
            "value":"_"
          }
      ]
```

</td><td>Yes</td><td>No default value</td></tr>
<tr><td>versionColumn</td><td>

The version of each HBase cell. You can use the current time, specific time, or a specific source column as the version. If you do not specify this parameter, the current time is used.

- index: the ID of a column in the source table, starting from 0. Make sure that the value can be converted to the LONG data type.
- type: the data type. If the data type is DATE, HBase Writer converts the date to the yyyy-MM-dd HH:mm:ss or yyyy-MM-dd HH:mm:ss SSS format. If you want to use the specified time as the version, set this parameter to -1.
- value: the specified time of the LONG data type.

The following code provides a configuration example:

```
"versionColumn":{
"index":1
}
```

```
"versionColumn":{
"index":-1,
"value":123456789
}
```

</td><td>No</td><td>No default value</td></tr>
<tr><td>nullMode</td><td>

The method used to process null values. Valid values:

- *skip*: HBase Writer does not write null values to HBase.
- *empty*: HBase Writer writes HConstants.EMPTY_BYTE_ARRAY (new byte [0]) to HBase instead of null values.

</td><td>No</td><td>*skip*</td></tr>
</table>

Parameter	Description	Required	Default value
walFlag	Specifies whether to enable write-ahead logging (WAL) for HBase. If you set this parameter to true, WAL is enabled. All the edits that are requested by an HBase client for all regions carried by the RegionServer are first recorded in the WAL log file (HLog). After the edits are recorded in the WAL log file, they are implemented to the MemStore, and a success notification is sent to the HBase client. If the edits fail to be recorded in the WAL log file, a failure notification is sent to the HBase client, and the edits are not implemented to the MemStore. If you set this parameter to false, WAL is disabled. This way, HBase Writer can write data more efficiently.	No	*false*
writeBufferSize	The write buffer size, in bytes, of the HBase client. If you specify this parameter, you must also specify the autoflush parameter. By default, the value of the autoflush parameter is false. autoflush: • If the value is true, the HBase client sends a PUT request each time it receives an edit. • If the value is false, the HBase client sends a PUT request only when its write buffer is full.	No	*8M*

Configure HBase Writer by using the codeless UI

This method is not supported.

Configure HBase Writer by using the code editor

In the following code, a synchronization node is configured to write data to HBase 1.1.X. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"hbase",// The writer type.
            "parameter":{
                "mode":"normal",// The write mode.
                "walFlag":"false",// WAL is disabled for HBase.
                "hbaseVersion":"094x",// The HBase version.
                "rowkeyColumn":[// The rowkey column of each row in the HBase table.
                    {
                        "index":"0",// The ID of a column in the source table.
                        "type":"string"// The data type.
                    },
                    {
                        "index":"-1",
                        "type":"string",
                        "value":"_"
                    }
                ],
                "nullMode":"skip",// The method used to process null values.
                "column":[// The names of the columns to which you want to write data.
                    {
                        "name":"columnFamilyName1:columnName1",// The name of a column in the HBase table.
                        "index":"0",// The ID of a column in the source table.
                        "index":"0",// The ID of a column in the source table.
                        "type":"string"// The data type.
                    },
                    {
                        "name":"columnFamilyName2:columnName2",
                        "index":"1",
                        "type":"string"
                    },
                    {
                        "name":"columnFamilyName3:columnName3",
                        "index":"2",
                        "type":"string"
                    }
                ],
                "encoding":"utf-8",// The encoding format.
                "table":"",// The name of the table to which you want to write data.
                "hbaseConfig":{// The properties of the HBase cluster, in the JSON format.
                    "hbase.zookeeper.quorum":"hostname",
                    "hbase.rootdir":"hdfs: //ip:port/database",
                    "hbase.cluster.distributed":"true"
                }
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.7. HBase11xsql Writer

This topic describes the parameters that are supported by HBase11xsql Writer and how to configure HBase11xsql Writer by using the codeless user interface (UI) and code editor.

HBase11xsql Writer writes large amounts of data to HBase tables that are created based on Phoenix. Phoenix can encode the primary key to rowkey. If you use the HBase API to write data to an HBase table that is created based on Phoenix, you must manually convert data, which is time-consuming and error-prone. However, HBase11xsql Writer writes data to HBase tables without manual data conversions.

HBase11xsql Writer connects to a remote HBase table by using Java Database Connectivity (JDBC), and executes an UPSERT statement to write data to the HBase table.

Notice HBase11xsql Writer supports only exclusive resource groups for Data Integration, but not the default resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Add a custom resource group for Data Integration.

Column order

The column order in the writer must match the column order in the reader. The column order in the reader defines the order of columns in each row of the output data. However, the column order in the writer is the expected order of columns in each row of the input data. Example:

Specified column order in the reader: c1, c2, c3, c4.

Specified column order in the writer: x1, x2, x3, x4.

In this case, the value of Column c1 in the reader is assigned to Column x1 in the writer. If the specified column order in the writer is x1, x2, x4, x3, the value of Column c3 is assigned to Column x4 and the value of Column c4 is assigned to Column x3.

Features

HBase11xsql Writer can write data of an indexed table to an HBase table and synchronously update all the indexed tables.

Limits

HBase11xsql Writer has the following limits:

- HBase11xsql Writer can write data only to HBase 1.x.
- HBase11xsql Writer can write data only to the tables that are created based on Phoenix but not to native HBase tables.
- HBase11xsql Writer cannot write data with timestamps.

How it works

HBase11xsql Writer connects to an HBase table by using the Phoenix JDBC driver, and executes an UPSERT statement to write large amounts of data to the table. Phoenix can synchronously update indexed tables when HBase11xsql Writer writes data to an HBase table.

Parameters

Parameter	Description	Required	Default value
plugin	The writer type. Set this parameter to hbase11xsql.	Yes	No default value
table	The name of the table to which you want to write data. The name is case-sensitive. In normal cases, the name of a table that is created based on Phoenix is all capitalized.	Yes	No default value
column	The names of the columns to which you want to write data. The name is case-sensitive. In normal cases, the name of each column in a table that is created based on Phoenix is all capitalized. Note • HBase11xsql Writer writes data in accordance with the order of the columns that are obtained from the reader. • You do not need to specify the data type for each column. HBase11xsql Writer automatically obtains the metadata of columns from Phoenix.	Yes	No default value

Parameter	Description	Required	Default value
hbaseConfig	The properties of the HBase cluster. The hbase.zookeeper.quorum parameter is required. It specifies the ZooKeeper ensemble servers. Note • Separate multiple IP addresses with commas (,), such as ip1,ip2,ip3. • The zookeeper.znode.parent parameter is optional. Default value: /hbase.	Yes	No default value
batchSize	The maximum number of data records to write at a time.	No	*256*
nullMode	The method to process null values. Valid values: • *skip*: HBase11xsql Writer does not write null values to the HBase table. • *empty*: HBase11xsql Writer writes 0 or an empty string instead of null values to the HBase table. For a column of the numeric type, HBase11xsql Writer writes 0. For a column of the VARCHAR type, HBase11xsql Writer writes an empty string.	No	*skip*

Configure HBase11xsql Writer by using the code editor

In the following code, a synchronization node is configured to write data to a HBase table by using the code editor. For more information, see Create a synchronization node by using the code editor.

```
{
  "type": "job",
  "version": "1.0",
  "configuration": {
    "setting": {
      "errorLimit": {
        "record": "0"
      },
      "speed": {
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"1"// The maximum transmission rate.
      }
    },
    "reader": {
      "plugin": "odps",
      "parameter": {
        "datasource": "",
        "table": "",
        "column": [],
        "partition": ""
      }
    },
    "plugin": "hbase11xsql",
    "parameter": {
      "table": "The name of the table to which you want to write data. The table name is case-sensitive.",
      "hbaseConfig": {
        "hbase.zookeeper.quorum": "The IP addresses of ZooKeeper ensemble servers of the destination HBase cluster. Obtain the IP addresses from product engineers (PEs).",
        "zookeeper.znode.parent": "The root znode of the destination HBase cluster. Obtain the znode information from PEs."
      },
      "column": [
        "columnName"
      ],
      "batchSize": 256,
      "nullMode": "skip"
    }
  }
}
```

FAQ

Q: What is the appropriate number of parallel threads? Can I increase the number of parallel threads to speed up the data synchronization?

A: The recommended number of parallel threads is 5 to 10. In the data import process, the default size of a Java virtual machine (JVM) heap is 2 GB. Parallel synchronization requires multiple threads. However, if excessive threads are run at the same time, data synchronization cannot speed up and the job performance may deteriorate due to frequent garbage collection (GC). We recommend that you set the number of parallel threads in the range of 5 to 10.

Q: What is the appropriate value for the batchSize parameter?

A: The default value of the batchSize parameter is 256. You can set the batchSize parameter based on the amount of data in each row. In most cases, each write operation writes 2 MB to 4 MB of data. You can set this parameter to the data volume of a write operation divided by the data volume of a row.

5.3.8. HDFS Writer

This topic describes the data types and parameters that are supported by HDFS Writer and how to configure HDFS Writer by using the codeless user interface (UI) and code editor.

HDFS Writer can write text, Optimized Row Columnar (ORC), or Parquet files to a specified directory in Hadoop Distributed File System (HDFS). You can associate the columns in the files with the columns in Hive tables. Before you configure HDFS Writer, you must configure a Hive data source. For more information, see Add a Hive data source.

ⓘ **Note** HDFS Writer supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration, Use the shared resource group, and Create a custom resource group for Data Integration.

Limits

- Alibaba Cloud Apsara File Storage for HDFS is not supported.
- HDFS Writer can write only text, ORC, and Parquet files that store logical two-dimensional tables to HDFS.
- HDFS is a distributed file system and does not have a schema. Therefore, you cannot write only data in some columns of a file to HDFS.
- Hive data types, such as DECIMAL, BINARY, ARRAYS, MAPS, STRUCTS, and UNION, are not supported.
- HDFS Writer can write data to only one partition in a partitioned Hive table at a time.
- To write a text file to HDFS, make sure that the delimiter in the file is the same as that in the Hive table that you want to associate with the file. This way, you can associate the columns in the file that is written to HDFS with those in the Hive table.
- You can use HDFS Writer in the environment in which Hive 1.1.1 and Hadoop 2.7.1 (JDK version: 1.7) are installed. JDK is short for Java Development Kit. HDFS Writer can write files to HDFS in test environments in which Hive 1.2.0 and Hadoop 2.5.0 or Hadoop 2.6.0 are installed.

How it works

HDFS Writer writes files to HDFS in the following way:

1. Creates a temporary directory that does not exist in HDFS based on the path parameter you specified.

   The temporary directory is specified in the format of path_Random suffix.

2. Writes files that are obtained from a reader to the temporary directory.
3. Moves the files from the temporary directory to the specified directory after all the files are written. The names of the files that you want to write to HDFS must be different from those of existing files in HDFS.
4. Deletes the temporary directory. If HDFS Writer fails to connect to HDFS due to a network interruption, you must manually delete the temporary directory and all the files in the temporary directory.

ⓘ **Note** To synchronize data, you must use an administrator account that has read and write permissions on the specific files.

Data types

HDFS Writer supports most Hive data types. Make sure that the data types of your system are supported.

The following table lists the Hive data types that are supported by HDFS Writer.

ⓘ **Note** The data types of the specified columns in the file must be the same as those of the columns in the Hive table.

Category	Hive data type
Integer	TINYINT, SMALLINT, INT, and BIGINT
Floating point	FLOAT and DOUBLE
String	CHAR, VARCHAR, and STRING
Boolean	BOOLEAN
Date and time	DATE and TIMESTAMP

Parameters

<table>
<tr><th>Parameter</th><th>Description</th><th>Required</th><th>Default value</th></tr>
<tr><td>datasource</td><td>The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.</td><td>Yes</td><td>No default value</td></tr>
<tr><td>defaultFS</td><td>The address of the NameNode node in HDFS, such as <code>hdfs://127.0.0.1:9000</code> . If you use the shared resource group for Data Integration, you cannot configure the advanced Hadoop parameters that are related to high availability. If you want to configure these parameters, you must use a custom resource group for Data Integration. For more information, see Create and use a custom resource group for Data Integration.</td><td>Yes</td><td>No default value</td></tr>
<tr><td>fileType</td><td>The format of the files that you want to write to HDFS. Valid values: <i>text</i>, <i>orc</i>, and <i>parquet</i>.
• <i>text</i>: a text file that maps a storage table in Hive
• <i>orc</i>: an ORC file that maps a compressed table in Hive
• <i>parquet</i>: a common Parquet file</td><td>Yes</td><td>No default value</td></tr>
<tr><td>path</td><td>The directory in HDFS to which you want to write files. HDFS Writer writes multiple files to the directory based on the configuration of parallel threads.
To associate the columns in a file with those in a Hive table, set the path parameter to the storage path of the Hive table in HDFS. For example, the storage path that is specified for the Hive data warehouse is <code>/user/hive/warehouse/</code> . In this case, the storage path of the hello table that is created in the test database is <code>/user/hive/warehouse/test.db/hello</code> .</td><td>Yes</td><td>No default value</td></tr>
<tr><td>fileName</td><td>The name prefix of the files that you want to write to HDFS. A random suffix is appended to the specified prefix to form the actual file name that is used by each thread.</td><td>Yes</td><td>No default value</td></tr>
<tr><td>column</td><td>The names of the columns to which you want to write data. You cannot write data only to some columns in the Hive table.
To associate the columns in a file with those in a Hive table, configure the name and type parameters for each column. The name parameter specifies the name of the column, and the type parameter specifies the data type of the column.
You can specify the column parameter in the following format:
<pre>
"column":
[
    {
        "name": "userName",
        "type": "string"
    },
    {
        "name": "age",
        "type": "long"
    }
]
</pre></td><td>Required if the fileType parameter is set to text or orc</td><td>No default value</td></tr>
</table>

Parameter	Description	Required	Default value
writeMode	The write mode. Valid values: - *append*: HDFS Writer writes the files based on the specified file name prefix and ensures that the actual file names do not conflict with the names of existing files. - *nonConflict*: HDFS Writer returns an error if a file with the specified file name prefix exists in the destination directory. - `truncate`: HDFS Writer deletes all existing files whose names start with the specified file name prefix from the destination directory before files are written to the directory. For example, if you set `fileName` to abc, all existing files whose names start with abc are deleted from the destination directory. **Note** Parquet files do not support the append mode. To write Parquet files, you must set the writeMode parameter to nonConflict.	Yes	No default value
fieldDelimiter	The column delimiter that is used in the files you want to write to HDFS. Make sure that you use the same delimiter as that in the Hive table. Otherwise, you cannot query data in the Hive table.	Required if the fileType parameter is set to text or orc	No default value
compress	The compression format of the files that you want to write to HDFS. By default, this parameter is left empty, which indicates that the files are not compressed. For a text file, the GZIP and BZIP2 compression formats are supported. For an ORC file, the Snappy compression format is supported. To compress an ORC file, you must install SnappyCodec. To install SnappyCodec,submit a ticket.	No	No default value
encoding	The encoding format of the files that you want to write to HDFS.	No	UTF-8

Parameter	Description	Required	Default value

parquetSchema	The schema of the Parquet files that you want to write to HDFS. This parameter is available only if the fileType parameter is set to parquet. Format:
	`message MessageTypeName {`
	`required, dataType, columnName;`
	`......................;`
	`}`
	Parameters:
	- MessageTypeName: the name of the MessageType object.
	- required: indicates that the column cannot be left empty. You can also specify optional based on your business requirements. We recommend that you specify optional for all columns.
	- dataType: Parquet files support various data types, such as BOOLEAN, INT32, INT64, INT96, FLOAT, DOUBLE, BINARY, and FIXED_LEN_BYTE_ARRAY. Set this parameter to BINARY if the column stores strings.
	Note Each line, including the last line, must end with a semicolon (;).
	Example:
	`message m {`
	`optional int64 id;`
	`optional int64 date_id;`
	`optional binary datetimestring;`
	`optional int32 dspId;`
	`optional int32 advertiserId;`
	`optional int32 status;`
	`optional int64 bidding_req_num;`
	`optional int64 imp;`
	`optional int64 click_num;`
	`}`
		No	No default value

Parameter	Description	Required	Default value

hadoopConfig	The settings of the advanced Hadoop parameters that are related to high availability. If you use the shared resource group for Data Integration, you cannot configure the advanced Hadoop parameters that are related to high availability. If you want to configure these parameters, you must use a custom resource group for Data Integration. For more information, see Create and use a custom resource group for Data Integration.
	`"hadoopConfig":{`
	`"dfs.nameservices": "testDfs",`
	`"dfs.ha.namenodes.testDfs": "namenode1,namenode2",`
	`"dfs.namenode.rpc-address.youkuDfs.namenode1": "",`
	`"dfs.namenode.rpc-address.youkuDfs.namenode2": "",`
	`"dfs.client.failover.proxy.provider.testDfs": "org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider"`
	`}`
		No	No default value

Parameter	Description	Required	Default value

dataxParquetMode	The synchronization mode for Parquet files. Valid values: fields and columns. If you set this parameter to fields, HDFS Writer can write data of complex data types, such as ARRAY, MAP, and STRUCT.
	If you set this parameter to fields, HDFS Writer supports HDFS over Object Storage Service (OSS). In this case, HDFS uses OSS as the storage service, and HDFS Writer writes Parquet files to OSS. You can add the following OSS-related parameters in the hadoopConfig parameter:
	• fs.oss.accessKeyId: the AccessKey ID of the account that you can use to connect to OSS
	• fs.oss.accessKeySecret: the AccessKey secret of the account that you can use to connect to OSS
	• fs.oss.endpoint: the endpoint of OSS
	The following sample code provides an example on how to connect to OSS:
	```json
	    "writer": {
	    "name": "hdfswriter",
	    "parameter": {
	        "defaultFS": "oss://test-bucket",
	        "fileType": "parquet",
	        "path": "/datasets/oss_demo/kpt",
	        "fileName": "test",
	        "writeMode": "truncate",
	        "compress": "SNAPPY",
	        "encoding": "UTF-8",
	        "hadoopConfig": {
	            "fs.oss.accessKeyId": "the-access-id",
	            "fs.oss.accessKeySecret": "the-access-key",
	            "fs.oss.endpoint": "oss-cn-hangzhou.aliyuncs.com"
	            },
	            "parquetSchema": "message test {\n    required int64 id;\n    optional binary name (UTF8);\n    optional int64 gmt_create;\n    required group map_col (MAP) {\n        repeated group key_value {\n            required binary key (UTF8);\n            required binary value (UTF8);\n        }\n    }\n    required group array_col (LIST) {\n        repeated group list {\n            required binary element (UTF8);\n        }\n    }\n    required group struct_col {\n        required int64 id;\n        required binary name (UTF8);\n    }\n}",
	            "dataxParquetMode": "fields"
	            }
	        }
	```
		No	*columns*

haveKerberos	Specifies whether Kerberos authentication is required. If you set this parameter to *true*, the kerberosKeytabFilePath and kerberosPrincipal parameters are required.	No	*false*

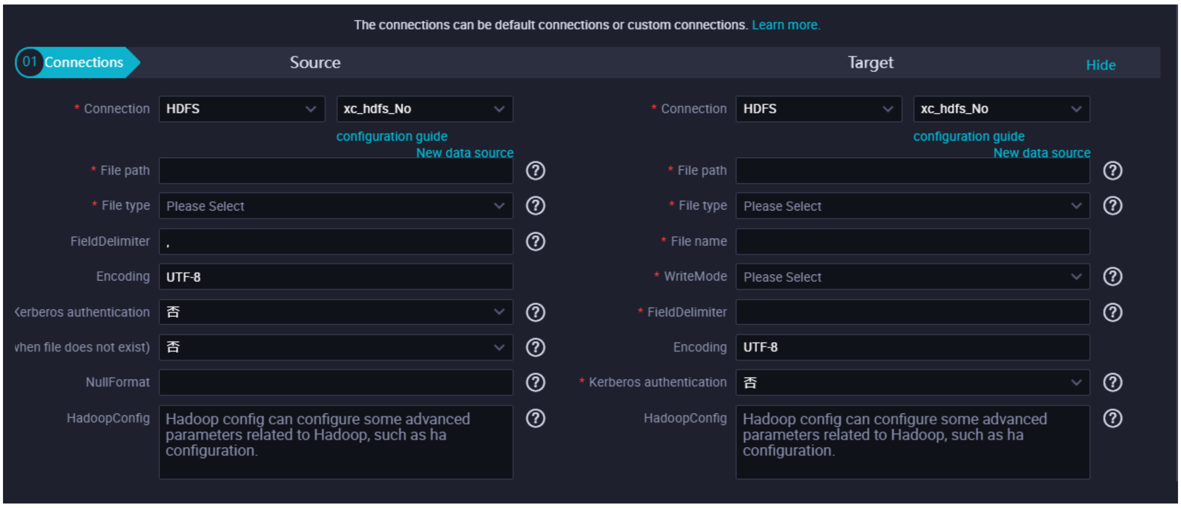
Parameter	Description	Required	Default value
kerberosKeytabFilePath	The absolute path of the keytab file for Kerberos authentication.	Required if the haveKerberos parameter is set to *true*	No default value

kerberosPrincipal	The Kerberos principal, such as ****/hadoopclient@**.***. This parameter is required if the haveKerberos parameter is set to *true*.
	The absolute path of the keytab file is required for Kerberos authentication. To use Kerberos authentication, you must configure Kerberos authentication on a custom resource group. The following code provides a configuration example:
	`"haveKerberos":true,`
	`"kerberosKeytabFilePath":"/opt/datax/**.keytab"`
	`,`
	`"kerberosPrincipal":"**/hadoopclient@**.**"`
		No	No default value

Configure HDFS Writer by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



The connections can be default connections or custom connections. Learn more.

01 Connections | Source | Target | Hide

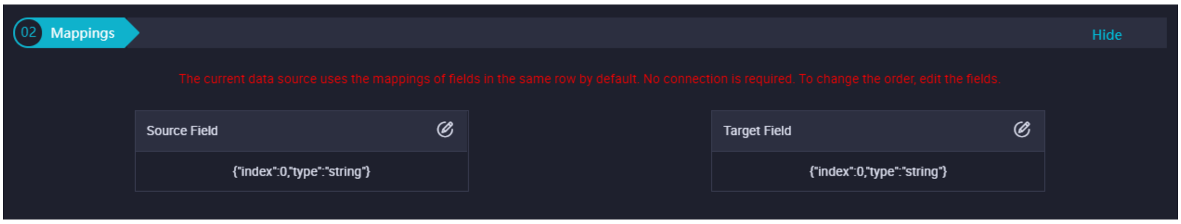
Source:
- * Connection: HDFS, xc_hdfs_No (configuration guide, New data source)
- * File path:
- * File type: Please Select
- FieldDelimiter: ,
- Encoding: UTF-8
- <erberos authentication: 否
- vhen file does not exist): 否
- NullFormat:
- HadoopConfig: Hadoop config can configure some advanced parameters related to Hadoop, such as ha configuration.

Target:
- * Connection: HDFS, xc_hdfs_No (configuration guide, New data source)
- * File path:
- * File type: Please Select
- * File name:
- * WriteMode: Please Select
- * FieldDelimiter:
- Encoding: UTF-8
- * Kerberos authentication: 否
- HadoopConfig: Hadoop config can configure some advanced parameters related to Hadoop, such as ha configuration.

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
File path	The directory in HDFS to which you want to write files. This parameter is equivalent to the path parameter that is described in the preceding section.
File type	The format of the files that you want to write to HDFS. This parameter is equivalent to the fileType parameter that is described in the preceding section. Valid values: *text*, *orc*, and *parquet*.
File name	The name prefix of the files that you want to write to HDFS. This parameter is equivalent to the fileName parameter that is described in the preceding section.

Parameter	Description
WriteMode	The write mode. This parameter is equivalent to the writeMode parameter that is described in the preceding section. Valid values: ○ append: HDFS Writer writes the files based on the specified file name prefix and ensures that the actual file names do not conflict with the names of existing files. ○ nonConflict: HDFS Writer returns an error if a file with the specified file name prefix exists in the destination directory. **Note** Parquet files do not support the append mode. They support only the nonConflict mode.
FieldDelimiter	The column delimiter that is used in the files you want to write to HDFS. This parameter is equivalent to the fieldDelimiter parameter that is described in the preceding section. Make sure that you use the same delimiter as that in the Hive table. Otherwise, you cannot query data in the Hive table.
Encoding	The encoding format. This parameter is equivalent to the encoding parameter that is described in the preceding section. Default value: UTF-8.
Kerberos authentication	Specifies whether Kerberos authentication is required. Default value: No. If you set this parameter to Yes, you must also specify the KeyTab file path and Principal Name parameters. For more information, see Configure Kerberos authentication.
HadoopConfig	The settings of the advanced Hadoop parameters that are related to high availability. If you use the shared resource group for Data Integration, you cannot configure the advanced Hadoop parameters that are related to high availability.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. By default, **Map Fields in the Same Line** is used to establish mappings between fields. You can click the icon to edit fields. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored. Make sure that the numbers of fields in the source and destination tables match.



The current data source uses the mappings of fields in the same row by default. No connection is required. To change the order, edit the fields.

{"index":0,"type":"string"}

{"index":0,"type":"string"}

3.

Configure HDFS Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to write data to HDFS. For more information about the parameters, see the preceding parameter description.

```
{
    "type": "job",
    "version": "2.0",// The version number.
    "steps": [
        {
            "stepType": "stream",
            "parameter": {},
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "hdfs",// The writer type.
            "parameter": {
                "path": "",// The directory in HDFS to which the files are written.
                "fileName": "",// The name prefix of the files that you want to write to HDFS.
                "compress": "",// The compression format of the files that you want to write to HDFS.
                "datasource": "",// The name of the data source.
                "column": [
                    {
                        "name": "col1",// The name of a column.
                        "type": "string"// The data type of a column.
                    },
                    {
                        "name": "col2",
                        "type": "int"
                    },
                    {
                        "name": "col3",
                        "type": "double"
                    },
                    {
                        "name": "col4",
                        "type": "boolean"
                    },
                    {
                        "name": "col5",
                        "type": "date"
                    }
                ],
                "writeMode": "",// The write mode.
                "fieldDelimiter": ",",// The column delimiter.
                "encoding": "",// The encoding format.
                "fileType": "text"// The format of the files that you want to write to HDFS.
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "setting": {
        "errorLimit": {
            "record": ""// The maximum number of dirty data records allowed.
        },
        "speed": {
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":3, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}
```

5.3.9. Memcache Writer

This topic describes the parameters that are supported by Memcache Writer and how to configure Memcache Writer by using the codeless user interface (UI) and code editor.

ApsaraDB for Memcache is a distributed in-memory database service with high performance, reliability, and scalability. ApsaraDB for Memcache is built on top of the Apsara distributed operating system and high-performance storage technologies, and provides a complete database solution that includes the hot standby, fault recovery, business monitoring, and data migration features.

ApsaraDB for Memcache is available right after an instance is created. It relieves the load on databases from dynamic websites and applications by caching data in the memory and therefore improves the response speed of websites and applications.

The following content describes the similarities and differences between ApsaraDB for Memcache databases and self-managed Memcached databases:

- ApsaraDB for Memcache databases are compatible with the Memcached protocol like self-managed Memcached databases. ApsaraDB for Memcache databases can be directly used in your environments.
- The difference is that the data, hardware infrastructure, network security, and system maintenance services used by ApsaraDB for Memcache databases are all deployed on the cloud. The pay-as-you-go billing method is used for these services.

Memcache Writer writes data to ApsaraDB for Memcache databases in compliance with the Memcached protocol.

Memcache Writer writes data only in the text format. The method of converting data types varies based on the format in which Memcache Writer writes data.

- text: Memcache Writer uses the specified column delimiter to serialize source data to a string.
- binary: This format is not supported.

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
writeMode	The write mode. Valid values: - *set*: stores the source data. - *add*: stores the source data only when its key does not exist in the destination ApsaraDB for Memcache database. This mode is not supported. - *replace*: uses the source data to replace the data record that has the same key as the source data in the destination ApsaraDB for Memcache database. This mode is not supported. - *append*: adds the source data to the end of an existing data record that has the same key as the source data, but does not update the expiration time of the existing data record. This mode is not supported. - *prepend*: adds the source data to the beginning of an existing data record that has the same key as the source data, but does not update the expiration time of the existing data record. This mode is not supported.	Yes	No default value

<table>
<tr><th>Parameter</th><th>Description</th><th>Required</th><th>Default value</th></tr>
<tr><td>writeFormat</td><td>The format in which Memcache Writer writes the source data. Only the text format is supported. text: serializes the source data to the text format. Memcache Writer uses the first column of the source data as the key and serializes the subsequent columns to strings by using the specified delimiter. Then, Memcache Writer writes key-value pairs to ApsaraDB for Memcache databases. Assume that the following source data exists: | ID	NAME	COUNT
23	"CDP"	100 | If you set the column delimiter to a backslash and a caret (\^), data is written to ApsaraDB for Memcache databases in the following format: | KEY (OCS)	VALUE(OCS)
---------	:----------
23	CDP\^100	</td><td>No</td><td>No default value</td></tr>
<tr><td>expireTime</td><td>The expiration time of the source data that is cached in ApsaraDB for Memcache databases. ApsaraDB for Memcache supports the expiration time in the following formats:
• unixtime: the UNIX timestamp, which indicates a specific point in time in the future when the data expires. The UNIX timestamp represents the number of seconds that have elapsed since 00:00:00 on January 1, 1970.
• seconds: the period of time in seconds starting from the current point in time. It specifies the period during which data is valid.
Note If the specified expiration time is greater than 30 days, the server identifies the time as a UNIX timestamp.</td><td>No</td><td><i>0</i>, which indicates that the data never expires.</td></tr>
<tr><td>batchSize</td><td>The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and ApsaraDB for Memcache and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.</td><td>No</td><td><i>1,024</i></td></tr>
</table>

Configure Memcache Writer by using the codeless UI

This method is not supported.

Configure Memcache Writer by using the code editor

In the following code, a synchronization node is configured to write data to an ApsaraDB for Memcache database by using the code editor. For more information, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"ocs",// The writer type.
            "parameter":{
                "writeFormat":"text",// The format in which Memcache Writer writes the source data.
                "expireTime":1000,// The expiration time of the source data that is cached in ApsaraDB for Memc
ache databases.
                "indexes":0,
                "datasource":"",// The name of the data source.
                "writeMode":"set",// The write mode.
                "batchSize":"256"// The number of data records to write at a time.
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.10. MongoDB Writer

This topic describes the data types and parameters that are supported by MongoDB Writer and how to configure MongoDB Writer by using the codeless user interface (UI) and code editor.

Background information

MongoDB Writer connects to a remote MongoDB database by using the Java client MongoClient and writes data to the database. The locking feature in the latest version of MongoDB is improved from database-level locking to document-level locking. This enables MongoDB Writer to efficiently write data to MongoDB databases. If you want to update data, specify the primary key.

Note

- Before you configure MongoDB Writer, you must configure a MongoDB data source. For more information, see Add a MongoDB data source.
- If you use ApsaraDB for MongoDB, a root account is provided for the MongoDB database by default.
- For security purposes, Data Integration can use only the account of a MongoDB database to connect to the MongoDB database. When you add a MongoDB data source, do not use the root account.

MongoDB Writer obtains data from a reader and converts the data from data types supported by Data Integration to data types supported by MongoDB. Data Integration does not support arrays. MongoDB supports arrays, and arrays support the indexing feature.

You can configure parameters to convert strings to MongoDB arrays. Then, MongoDB Writer uses parallel threads to write the arrays to a MongoDB database.

Data types

MongoDB Writer supports most MongoDB data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by MongoDB Writer.

Category	MongoDB data type
Integer	INT and LONG
Floating point	DOUBLE
String	STRING and ARRAY
Date and time	DATE
Boolean	BOOLEAN
Binary	BYTES

Note When MongoDB Writer writes data of the DATE data type to a MongoDB database, MongoDB Writer converts the data to the DATETIME data type.

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
collectionName	The name of the collection in MongoDB.	Yes	No default value
column	The names of the document fields to which you want to write data. Specify the names in an array. • name: the name of a field. • type: the data type of a field. • splitter: the delimiter. Configure this parameter only if you want to convert strings to arrays.	Yes	No default value

Parameter	Description	Required	Default value
writeMode	The write mode. The following parameters are included: • *isReplace*: If you set isReplace to true, MongoDB Writer overwrites the data that contains the same primary key in the destination table. If you set isReplace to false, MongoDB Writer does not overwrite the data. • *replaceKey*: the primary key for each data record. Data is overwritten based on the primary key. The primary key must be unique.	No	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to delete outdated data. If the preSql parameter is left empty, no SQL statement is executed before the synchronization node is run. Make sure that the value of the preSql parameter is specified based on the JSON syntax.	No	No default value

Before the synchronization node is run, Data Integration executes the SQL statement specified by the preSql parameter. Then, Data Integration starts to write data. The preSql parameter does not affect the data that is written. You can configure the preSql parameter to ensure the idempotence of the write operation. For example, you can configure the preSql parameter to delete outdated data before a synchronization node is run based on your business requirements. If the synchronization node fails, you need to only rerun the synchronization node.

Requirements on the format of the preSql parameter:

- Configure the type parameter to specify the action type. Valid values: drop and remove. Example: `"preSql":{"type":"remove"}`.
  - *drop*: deletes the collection specified by the collectionName parameter and the data in the collection.
  - *remove*: deletes data based on specified conditions.
  - *json*: the conditions used to delete data. Example: `"preSql":{"type":"remove", "json":"{'operationTime':{'$gte':ISODate('${last_day}T00:00:00.424+0800')}}"}`. `${last_day}` is a scheduling parameter of DataWorks. You can configure this parameter in the format of `$[yyyy-mm-dd]`. Other operators and functions are also supported, such as comparison operators $gt, $lt, $gte, and $lte, logical operators $and and $or, and functions max, min, sum, avg, and ISODate. You can use them based on your business requirements.

    Data Integration uses the following standard MongoDB API to query and delete the specified data:

    ```
    query=(BasicDBObject) com.mongodb.util.JSON.parse(json);
    col.deleteMany(query);
    ```

    > **Note** If you want to delete data based on conditions, we recommend that you specify the conditions in the JSON format.

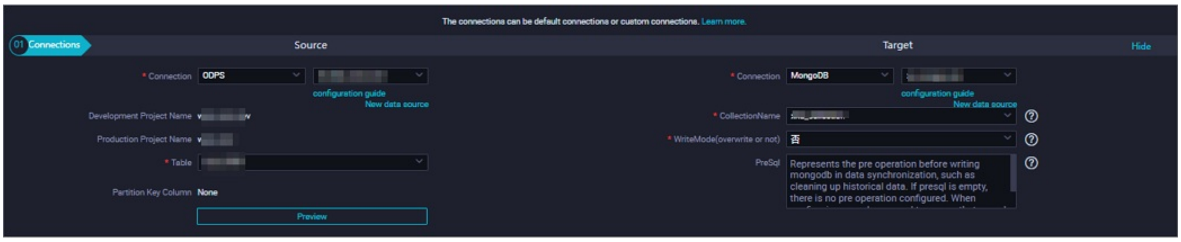
  - *item*: the name, condition, and value for filtering data. Example: `"preSql":{"type":"remove","item":[{"name":"pv","value":"100","condition":"$gt"},{"name":"pid","value":"10"}]}`.

    Data Integration configures query conditions based on the value of the item parameter and deletes data by using the standard MongoDB API. Example: `col.deleteMany(query);`.
- If the value of the preSql parameter cannot be recognized, no SQL statement is executed.

Configure MongoDB Writer by using the codeless UI

1. Configure data sources.

Configure **Source** and **Target** for the synchronization node.



The connections can be default connections or custom connections. Learn more.

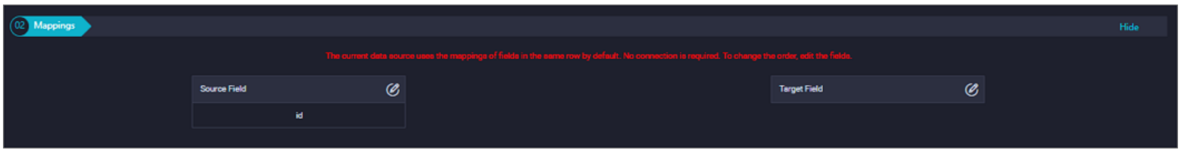
01 Connections — Source — Target — Hide

Source:
- * Connection: ODPS [redacted]
- configuration guide
- New data source
- Development Project Name: [redacted]
- Production Project Name: [redacted]
- * Table: [redacted]
- Partition Key Column: None
- Preview

Target:
- * Connection: MongoDB [redacted]
- configuration guide
- New data source
- * CollectionName: [redacted]
- * WriteMode(overwrite or not): 否
- PreSql: Represents the pre operation before writing mongodb in data synchronization, such as cleaning up historical data. If presql is empty, there is no pre operation configured. When

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
CollectionName	The name of the collection in MongoDB. This parameter is equivalent to the collectionName parameter that is described in the preceding section.
WriteMode(overwrite or not)	The write mode. This parameter is equivalent to the writeMode parameter that is described in the preceding section.
PreSql	The SQL statement that you want to execute before the synchronization node is run. This parameter is equivalent to the preSql parameter that is described in the preceding section. For example, you can set this parameter to the SQL statement that is used to delete outdated data. If the preSql parameter is left empty, no SQL statement is executed before the synchronization node is run. Make sure that the value of the preSql parameter is specified based on the JSON syntax.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. By default, the system maps the field in a row of the source to the field in the same row of the destination. You can click the ✎ icon to manually edit fields in the destination.



02 Mappings — Hide

The current data source uses the mappings of fields in the same row by default. No connection is required. To change the order, edit the fields.

Source Field	Target Field
id	

3.

Configure MongoDB Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to write data to MongoDB. For more information about the parameters, see the preceding parameter description.

```
{
    "type": "job",
    "version": "2.0",// The version number.
    "steps": [
        {
            "stepType": "stream",
            "parameter": {},
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "mongodb",// The writer type.
            "parameter": {
                "datasource": "",// The name of the data source.
                "column": [
                    {
                        "name": "_id",// The name of the field.
                        "type": "ObjectId"// The data type of the field. If you set the replaceKey parameter to
                        "type": "ObjectId"// The data type of the field. If you set the replaceKey parameter to
_id, you must set the type parameter to ObjectId. If you set the type parameter to string, the data cannot be o
verwritten.
                    },
                    {
                        "name": "age",
                        "type": "int"
                    },
                    {
                        "name": "id",
                        "type": "long"
                    },
                    {
                        "name": "wealth",
                        "type": "double"
                    },
                    {
                        "name": "hobby",
                        "type": "array",
                        "splitter": " "
                    },
                    {
                        "name": "valid",
                        "type": "boolean"
                    },
                    {
                        "name": "date_of_join",
                        "format": "yyyy-MM-dd HH:mm:ss",
                        "type": "date"
                    }
                ],
                "writeMode": {// The write mode.
                    "isReplace": "true",
                    "replaceKey": "_id"
                },
                "collectionName": "datax_test"// The name of the collection.
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "setting": {
        "errorLimit": {// The maximum number of dirty data records allowed.
            "record": "0"
        },
        "speed": {
            "throttle": true,// Specifies whether to enable bandwidth throttling. The value false indicates tha
t bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps
parameter takes effect only when the throttle parameter is set to true.
            "concurrent": 1,// The maximum number of parallel threads.
            "mbps": "1"// The maximum transmission rate.
        },
       "jvmOption": "-Xms1024m -Xmx1024m"
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}
```

5.3.11. MySQL Writer

This topic describes the data types and parameters that are supported by MySQL Writer and how to configure MySQL Writer by using the codeless user interface (UI) and code editor.

Prerequisites

Before you configure MySQL Writer, you must configure a MySQL data source. For more information, see Add a MySQL data source.

Background information

MySQL Writer writes data to tables stored in MySQL databases. MySQL Writer connects to a remote MySQL database by using Java Database Connectivity (JDBC), and executes an `INSERT INTO` or a `REPLACE INTO` statement to write data to the MySQL database. MySQL uses the InnoDB engine so that data is written to the database in batches.

MySQL Writer can also be used as a migration tool by users such as database administrators. MySQL Writer obtains data from a reader and writes the data to the destination database based on value of the writeMode parameter.

> ? **Note** A synchronization node that uses MySQL Writer must have at least the permissions to execute the `INSERT INTO or REPLACE INTO` statement. Whether other permissions are required depends on the SQL statements that you specify in the preSql and postSql parameters when you configure the node.

Data types

MySQL Writer supports most MySQL data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by MySQL Writer.

Category	MySQL data type
Integer	INT, TINYINT, SMALLINT, MEDIUMINT, BIGINT, and YEAR
Floating point	FLOAT, DOUBLE, and DECIMAL
String	VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT
Date and time	DATE, DATETIME, TIMESTAMP, and TIME
Boolean	BOOL
Binary	TINYBLOB, MEDIUMBLOB, BLOB, LONGBLOB, and VARBINARY

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table to which you want to write data.	Yes	No default value

Parameter	Description	Required	Default value
writeMode	The write mode. Valid values: *insert into*, *on duplicate key update*, and *replace into*. - *insert into*: If a primary key conflict or unique index conflict occurs, data cannot be written to the conflicting rows, and the data that is not written to these rows is regarded as dirty data. If you create a synchronization node by using the code editor, set writeMode to *insert*. - *on duplicate key update*: If no primary key conflict or unique index conflict occurs, the data is processed in the same way as that when you set this parameter to `insert into`. If a conflict occurs, specified fields in the original rows are replaced by new rows, and data is written to MySQL. If you create a synchronization node by using the code editor, set writeMode to *update*. - *replace into*: If no primary key conflict or unique index conflict occurs, the data is processed in the same way as that when you set this parameter to `insert into`. If a conflict occurs, the original rows are deleted, and new rows are inserted. This indicates that all fields of the original rows are replaced. If you create a synchronization node by using the code editor, set writeMode to *replace*.	No	*insert into*
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as `"column": ["id", "name", "age"]`. If you want to write data to all the columns in the destination table, set this parameter to an asterisk (*), such as `"column": ["*"]`.	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor. For example, you can set this parameter to the TRUNCATE TABLE tablename statement to delete outdated data. **Note** If you specify multiple SQL statements, the statements are not executed in the same transaction.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor. For example, you can set this parameter to the `ALTER TABLE tablename add colname timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP` statement to add a timestamp. **Note** If you specify multiple SQL statements, the statements are not executed in the same transaction.	No	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and MySQL and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

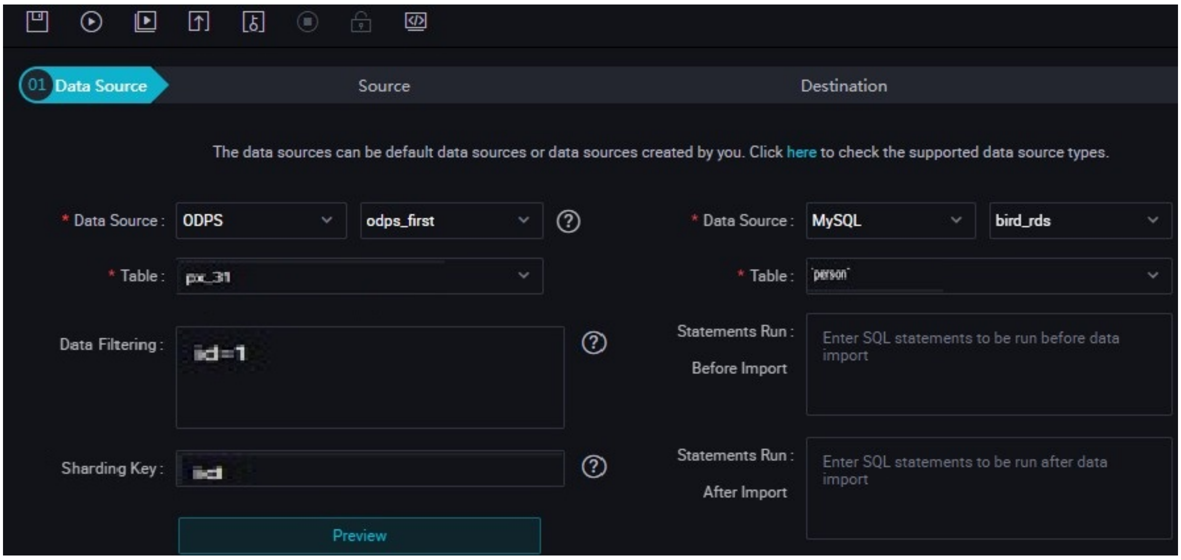
Configure MySQL Writer by using the codeless UI

Create a synchronization node and configure the node. For more information, see Configure a synchronization node by using the codeless UI.

Perform the following steps on the configuration tab of the node:

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



01 Data Source | Source | Destination

The data sources can be default data sources or data sources created by you. Click here to check the supported data source types.

* Data Source : ODPS | odps_first

* Table : px_31

Data Filtering : id=1

Sharding Key : id

Preview

* Data Source : MySQL | bird_rds

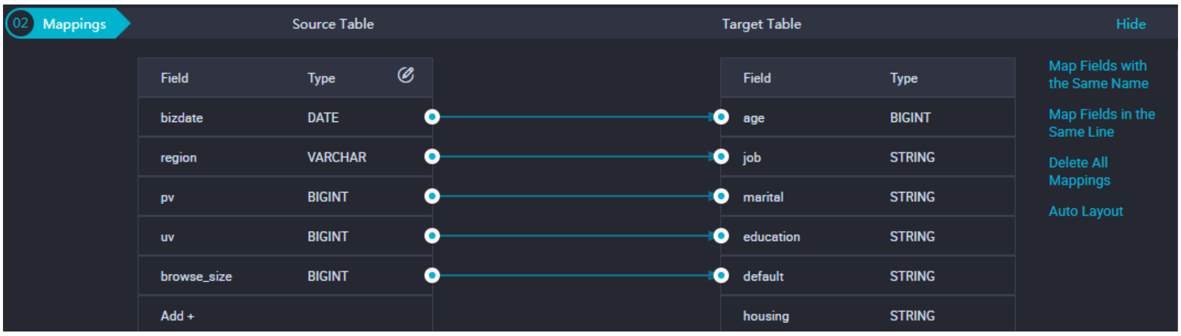
* Table : `person`

Statements Run Before Import : Enter SQL statements to be run before data import

Statements Run After Import : Enter SQL statements to be run after data import

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The table to which you want to write data. This parameter is equivalent to the table parameter that is described in the preceding section.
Statement Run Before Writing	The SQL statement that you want to execute before the synchronization node is run. This parameter is equivalent to the preSql parameter that is described in the preceding section.
Statement Run After Writing	The SQL statement that you want to execute after the synchronization node is run. This parameter is equivalent to the postSql parameter that is described in the preceding section.
Solution to Primary Key Violation	The write mode. This parameter is equivalent to the writeMode parameter that is described in the preceding section.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.



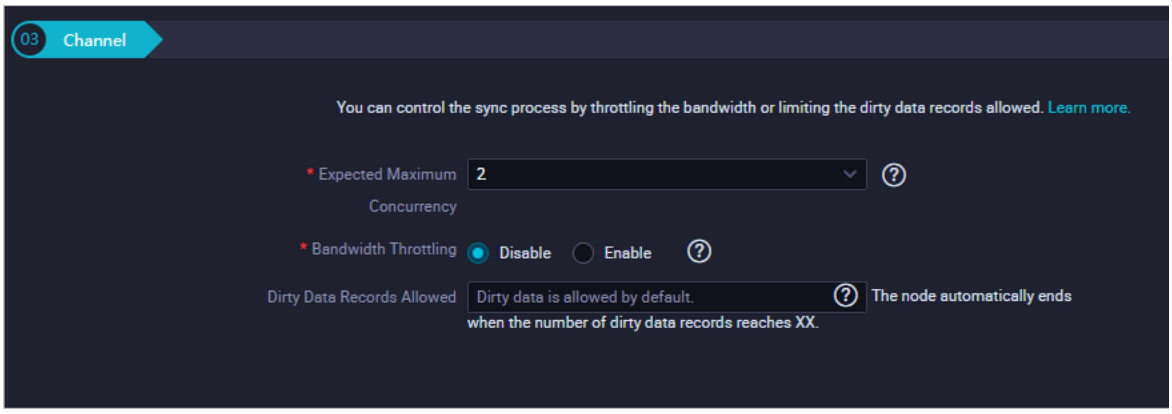
02 Mappings | Source Table | Target Table | Hide

Field	Type	Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
Add +		housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.

Operation	Description
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3. Configure channel control policies.



03 Channel

You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: Dirty data is allowed by default. The node automatically ends when the number of dirty data records reaches XX.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.

4.

Configure MySQL Writer by using the code editor

In the following code, a synchronization node is configured to write data to a MySQL database by using the code editor. For more information, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"mysql",// The writer type.
            "parameter":{
                "postSql":[],// The SQL statement that you want to execute after the synchronization node is run.
                "datasource":"",// The name of the data source.
                "column":[// The names of the columns to which you want to write data.
                    "id",
                    "value"
                ],
                "writeMode":"insert",// The write mode. Valid values: insert, replace, and update.
                "batchSize":1024,// The number of data records to write at a time.
                "table":"",// The name of the table to which you want to write data.
                "preSql":[
                     "delete from XXX;" // The SQL statement that you want to execute before the synchronization node is run.
                   ]
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{// The maximum number of dirty data records allowed.
            "record":"0"
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.12. Oracle Writer

This topic describes the data types and parameters that are supported by Oracle Writer and how to configure Oracle Writer by using the codeless user interface (UI) and code editor.

Oracle Writer writes data to tables in primary Oracle databases. Oracle Writer connects to a remote Oracle database by using Java Database Connectivity (JDBC) and executes the `INSERT INTO` statement to write data to the Oracle database.

Note

- Before you configure Oracle Writer, you must add an Oracle data source. For more information, see Add an Oracle data source.
- Oracle Writer uses the ojdbc6-12.1.1.jar driver. For more information about the supported versions of Oracle JDBC drivers, see Oracle JDBC FAQ.

Oracle Writer is designed for extract, transform, load (ETL) developers to import data in data warehouses to Oracle databases. Oracle Writer can also be used as a data migration tool by users such as database administrators.

Oracle Writer obtains data from a reader, connects to a remote Oracle database by using JDBC, and then executes an SQL statement to write data to the Oracle database.

Data types

Oracle Writer supports most Oracle data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by Oracle Writer.

Category	Oracle data type
Integer	NUMBER, ROWID, INTEGER, INT, and SMALLINT
Floating point	NUMERIC, DECIMAL, FLOAT, DOUBLE PRECISION, and REAL
String	LONG, CHAR, NCHAR, VARCHAR, VARCHAR2, NVARCHAR2, CLOB, NCLOB, CHARACTER, CHARACTER VARYING, CHAR VARYING, NATIONAL CHARACTER, NATIONAL CHAR, NATIONAL CHARACTER VARYING, NATIONAL CHAR VARYING, and NCHAR VARYING
Date and time	TIMESTAMP and DATE
Boolean	BIT and BOOLEAN
Binary	BLOB, BFILE, RAW, and LONG RAW

Parameters

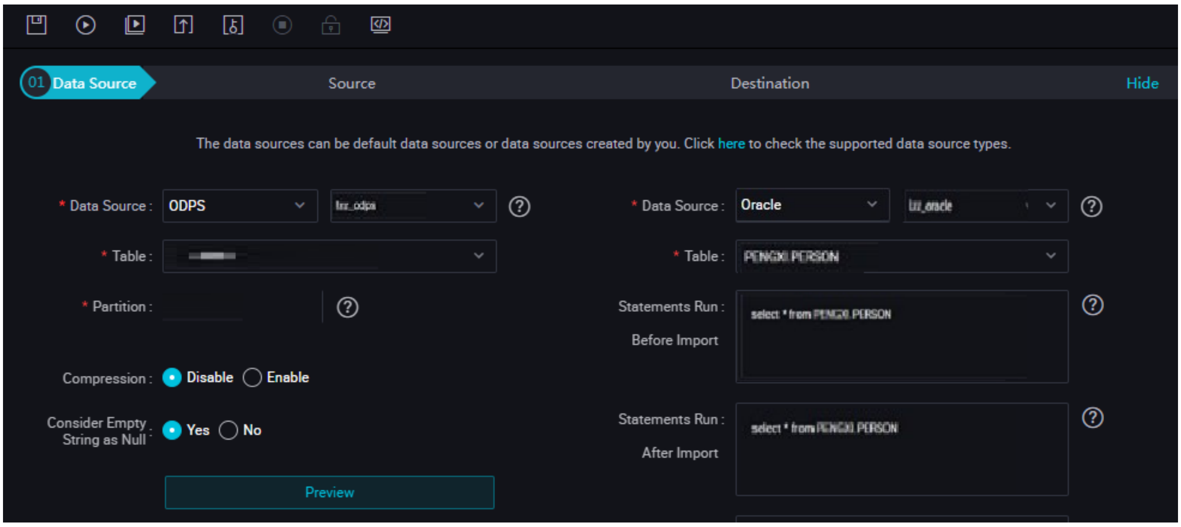
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table to which you want to write data. If the table uses the default schema for the destination database, you must set this parameter to the name of the table to which you want to write data. If the table uses a custom schema, you must specify this parameter in the Schema name.Name of the table to which you want to write data format.	Yes	No default value
writeMode	The write mode. Valid value: *insert into*. If a primary key conflict or unique index conflict occurs, data cannot be written to the conflicting rows, and the data that is not written to these rows is regarded as dirty data.	No	*insert into*
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as `"column": ["id","name","age"]`. If you want to write data to all the columns in the destination table, set this parameter to an asterisk (*), such as `"column":["*"]`.	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to delete outdated data. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value

Parameter	Description	Required	Default value
postSql	The SQL statement that you want to execute after the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to add a timestamp. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and Oracle and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

Configure Oracle Writer by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



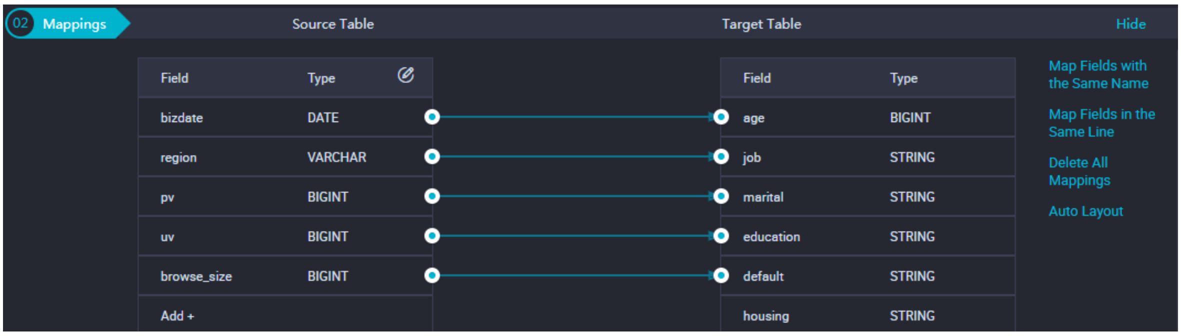
The data sources can be default data sources or data sources created by you. Click here to check the supported data source types.

Source:
- * Data Source : ODPS lxx_odps
- * Table : [illegible]
- * Partition :
- Compression : ● Disable ○ Enable
- Consider Empty String as Null : ● Yes ○ No

Destination:
- * Data Source : Oracle lxx_oracle
- * Table : PENGXI.PERSON
- Statements Run Before Import : select * from PENGXI.PERSON
- Statements Run After Import : select * from PENGXI.PERSON

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The table to which you want to write data. This parameter is equivalent to the table parameter that is described in the preceding section.
Statement Run Before Writing	The SQL statement that you want to execute before the synchronization node is run. This parameter is equivalent to the preSql parameter that is described in the preceding section.
Statement Run After Writing	The SQL statement that you want to execute after the synchronization node is run. This parameter is equivalent to the postSql parameter that is described in the preceding section.

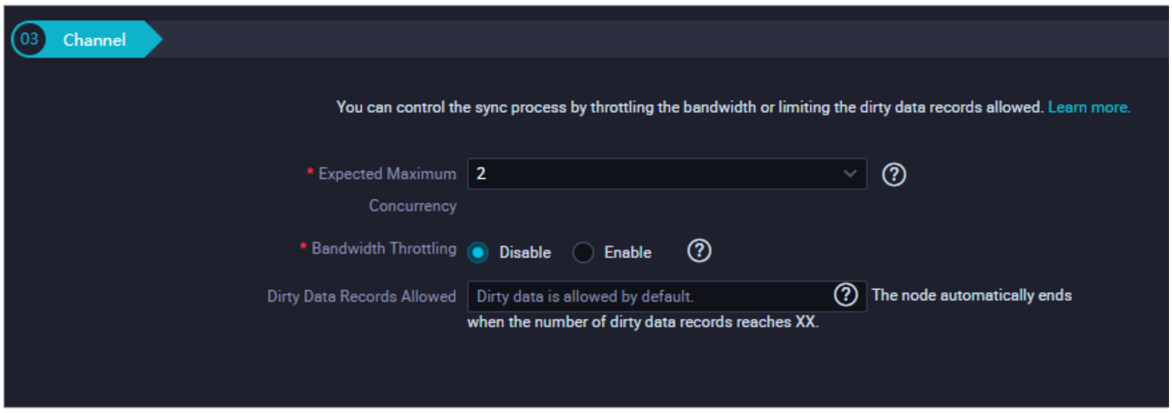
2. Configure field mapping. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.



Field	Type	Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
Add +		housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: Dirty data is allowed by default. The node automatically ends when the number of dirty data records reaches XX.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.

4.

Configure Oracle Writer by using the code editor

In the following code, a synchronization node is configured to write data to an Oracle database. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

Notice Delete the comments from the following code before you run the code:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"oracle",// The writer type.
            "parameter":{
                "postSql":[],// The SQL statement that you want to execute after the synchronization node is ru
n.
                "datasource":"",
                "session":[],// The settings of the session to the database.
                "column":[// The names of the columns to which you want to write data.
                    "id",
                    "name"
                ],
                "encoding":"UTF-8",// The encoding format.
                "batchSize":1024,// The number of data records to write at a time.
                "table":"",// The name of the table to which you want to write data.
                "preSql":[]// The SQL statement that you want to execute before the synchronization node is run
.
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.13. OSS Writer

This topic describes the parameters that are supported by OSS Writer and how to configure OSS Writer by using the codeless user interface (UI) and code editor.

Background information

OSS Writer writes one or more CSV-like files to Object Storage Service (OSS). The number of files that are written to OSS depends on the number of parallel threads and the total number of files that you want to write to OSS.

Note Before you configure OSS Writer, you must configure an OSS data source. For more information, see Add an OSS data source.

OSS Writer can write files that store logical two-dimensional tables, such as CSV files that store text data, to OSS. For more information about OSS, see What is OSS?

OSS stores only unstructured data. Therefore, OSS Writer converts the data obtained from a reader to text files and writes the files to OSS. OSS Writer provides the following features:

- Writes only text files to OSS. The data in the files must be organized as logical two-dimensional tables.
- Writes CSV-like files with custom delimiters to OSS.
- Uses parallel threads to write files to OSS. Each thread writes one file to OSS.
- Supports object rotation. Files are written to OSS as objects. If the size of a file exceeds a specific threshold, OSS Writer writes excess data as another object.

OSS Writer does not support the following features:

- Uses parallel threads to write a single file to an FTP server.
- Distinguishes between data types. OSS does not distinguish between data types. OSS Writer writes all data as strings to OSS.

Parameters

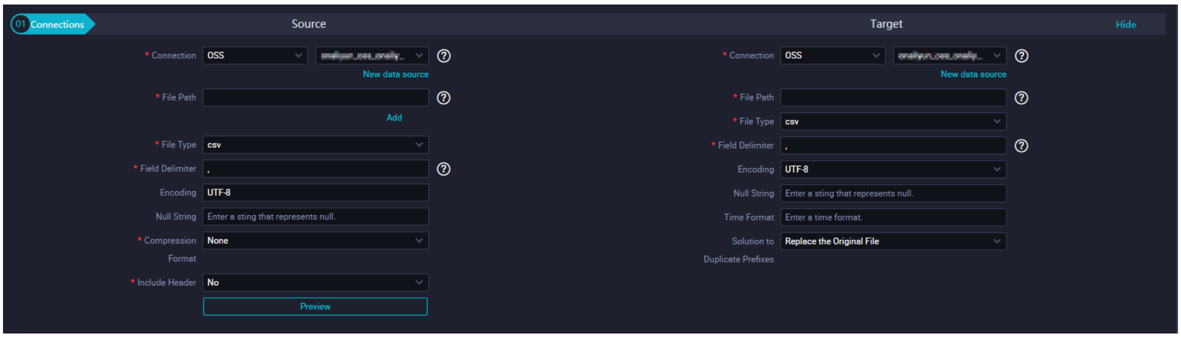
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
object	The prefix for the names of the files that you want to write to OSS. OSS simulates the directory effect by adding delimiters to file names. Examples: • If you set the object parameter to `datax`, the names of the files start with datax and end with random strings. • If you set the object parameter to `cdo/datax`, the names of the files start with `/cdo/datax` and end with random strings. OSS uses forward slashes (/) in file names to simulate the directory effect. If you do not want to add a random universally unique identifier (UUID) as the suffix, we recommend that you set the `writeSingleObject` parameter to true. For more information, see the description of the writeSingleObject parameter.	Yes	No default value
writeMode	The write mode. Valid values: • *truncate*: OSS Writer deletes all existing objects whose names start with the specified prefix before it writes files to OSS. For example, if you set the object parameter to `abc`, OSS Writer deletes all the objects whose names start with abc before it writes files to OSS. • *append*: OSS Writer writes all files to OSS and suffixes the file names with random UUIDs to ensure that the names of the files are different from the names of existing objects. For example, if you set the object parameter to DI, the actual names of the files written to OSS are in the DI_****_****_**** format. • *nonConflict*: If OSS contains objects whose names start with the specified prefix, OSS Writer returns an error. For example, if you set the `object` parameter to abc and OSS contains an object named abc123, OSS Writer returns an error.	Yes	No default value
writeSingleObject	Specifies whether to write a single file to OSS at a time. Valid values: • *true*: OSS Writer writes a single file to OSS at a time. • *false*: OSS Writer writes multiple files to OSS at a time.	No	*false*

Parameter	Description	Required	Default value
fileFormat	The format in which OSS Writer writes files to OSS. Valid values: - *csv*: If a file is written as a CSV file, the file must follow CSV specifications. If the data in the file contains column delimiters, the column delimiters are escaped by double quotation marks ("). - *text*: If a file is written as a text file, the data in the file is separated by column delimiters. In this case, OSS Writer does not escape the column delimiters. **Note** OSS Writer can write Parquet files to OSS. If you want to write Parquet files to OSS, you must configure the parquetschema parameter to define the related data type.	No	*text*
compress	The compression type of the files that you want to write to OSS. This parameter is available only in the code editor. **Note** CSV and text files cannot be compressed. Parquet and ORC files can be compressed in a format such as Snappy and ZIP.	No	No default value
fieldDelimiter	The column delimiter that is used in the files that you want to write to OSS.	No	*,*
encoding	The encoding format of the files that you want to write to OSS.	No	*utf-8*
nullFormat	The string that represents a null pointer. No standard strings can represent a null pointer in text files. You can use this parameter to define a string that represents a null pointer. For example, if you set `nullFormat` to `null`, Data Integration considers null as a null pointer.	No	No default value
header	The table headers in the files that you want to write to OSS. Example: `["id", "name", "age"]`.	No	No default value
maxFileSize (advanced parameter, which is available only in the code editor)	The maximum size of a single file that can be written to OSS. Default value: 100000. Unit: MB. OSS Writer performs object rotation based on the value of this parameter. Object rotation is similar to log rotation of Log4j. When a file is uploaded to OSS in multiple parts, the maximum size of a part is 10 MB. This size is the minimum granularity used for object rotation. If you set the maxFileSize parameter to a value that is less than 10 MB, the maximum size of a single file that can be written to OSS is still 10 MB. The InitiateMultipartUploadRequest operation can be used to upload a file in a maximum of 10,000 parts at a time. If object rotation occurs, suffixes, such as _1, _2, and _3, are appended to the new object names that consist of prefixes and random UUIDs.	No	100,000MB
suffix (advanced parameter, which is available only in the code editor)	The file name extension of the files that you want to write to OSS. For example, if you set the suffix parameter to *.csv*, the final name of a file written to OSS is in the fileName****.csv format.	No	No default value

Configure OSS Writer by using the codeless UI

1. Configure data sources.

Configure **Source** and **Target** for the synchronization node.



01 Connections

Source

- * Connection: OSS, [redacted]
- New data source
- * File Path:
- Add
- * File Type: csv
- * Field Delimiter: ,
- Encoding: UTF-8
- Null String: Enter a sting that represents null.
- * Compression Format: None
- * Include Header: No
- Preview

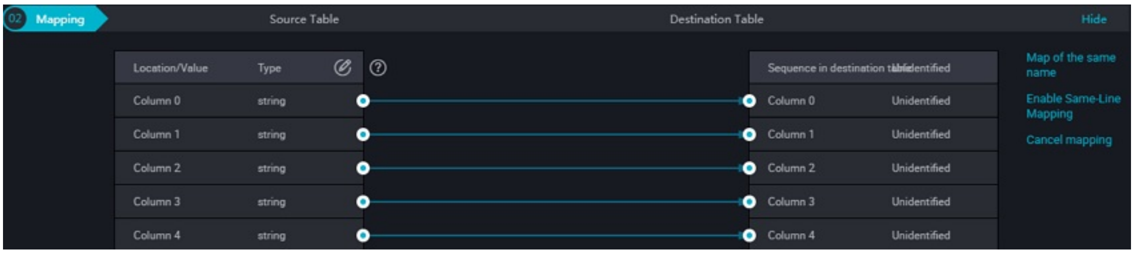
Target

- * Connection: OSS, [redacted]
- New data source
- * File Path:
- * File Type: csv
- * Field Delimiter: ,
- Encoding: UTF-8
- Null String: Enter a sting that represents null.
- Time Format: Enter a time format.
- Solution to Duplicate Prefixes: Replace the Original File

Hide

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Object Name (Path Included)	The prefix for the names of the files that you want to write to OSS. This parameter is equivalent to the object parameter that is described in the preceding section. Do not use this parameter to specify the name of your OSS bucket.
File Type	The format in which OSS Writer writes the files to OSS. Valid values: *csv*, *text*, and parquet.
Field Delimiter	This parameter is equivalent to the fieldDelimiter parameter that is described in the preceding section. By default, a comma (,) is used as a column delimiter.
Encoding	This parameter is equivalent to the encoding parameter that is described in the preceding section. Default value: *UTF-8*.
Null String	The string that represents a null pointer. This parameter is equivalent to the nullFormat parameter that is described in the preceding section. If the data in the file that you want to write contains the specified string, the string is replaced with null.
Time Format	The format in which data of the DATE data type is serialized in objects. Example: `yyyy-MM-dd`.
Solution to Duplicate Prefixes	The solution to prefix conflicts. If the prefix of the name of an existing object is the same as the specified prefix, OSS Writer replaces the existing object with the new object, appends data to the existing object, or returns an error.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.



02 Mapping

Source Table: Location/Value	Type	Destination Table: Sequence in destination table	Unidentified
Column 0	string	Column 0	Unidentified
Column 1	string	Column 1	Unidentified
Column 2	string	Column 2	Unidentified
Column 3	string	Column 3	Unidentified
Column 4	string	Column 4	Unidentified

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.

3.

Configure OSS Writer by using the code editor

In the following code, a synchronization node is configured to write data to OSS. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"oss",// The writer type.
            "parameter":{
                "nullFormat":"",// The string that represents a null pointer.
                "dateFormat":"",// The format in which data of the DATE data type is serialized in objects.
                "datasource":"",// The name of the data source.
                "writeMode":"",// The write mode.
                "writeSingleObject":"false", // Specifies whether to write a single file to OSS at a time.
                "encoding":"",// The encoding format.
                "fieldDelimiter":","// The column delimiter.
                "fileFormat":"",// The format in which OSS Writer writes files to OSS.
                "object":""// The prefix for the names of the files that you want to write to OSS.
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

Write ORC or Parquet files to OSS

OSS Writer writes ORC or Parquet files to OSS in the way in which HDFS Writer writes data to Hadoop Distributed File System (HDFS). In addition to the parameters for OSS Writer, extended parameters, such as Path and FileFormat, are added for OSS Writer. For more information about the extended parameters, see HDFS Writer.

The following sample code provides examples on how to configure a synchronization node to write an ORC file to OSS and how to configure a synchronization node to write a Parquet file to OSS.

- Write an ORC file to OSS

```
{
      "stepType": "oss",
      "parameter": {
        "datasource": "",
        "fileFormat": "orc",
        "path": "/tests/case61",
        "fileName": "orc",
        "writeMode": "append",
        "column": [
          {
            "name": "col1",
            "type": "BIGINT"
          },
          {
            "name": "col2",
            "type": "DOUBLE"
          },
          {
            "name": "col3",
            "type": "STRING"
          }
        ],
        "writeMode": "append",
        "fieldDelimiter": "\t",
        "compress": "NONE",
        "encoding": "UTF-8"
      }
    }
```

- Write a Parquet file to OSS

```
{
      "stepType": "oss",
      "parameter": {
        "datasource": "",
        "fileFormat": "parquet",
        "path": "/tests/case61",
        "fileName": "test",
        "writeMode": "append",
        "fieldDelimiter": "\t",
        "compress": "SNAPPY",
        "encoding": "UTF-8",
        "parquetSchema": "message test { required int64 int64_col;\n required binary str_col (UTF8);\nrequire
d group params (MAP) {\nrepeated group key_value {\nrequired binary key (UTF8);\nrequired binary value (UTF8)
;\n}\n}\nrequired group params_arr (LIST) {\n  repeated group list {\n    required binary element (UTF8);\n
}\n}\nrequired group params_struct {\n  required int64 id;\n required binary name (UTF8);\n }\nrequired group
params_arr_complex (LIST) {\n  repeated group list {\n    required group element {\n required int64 id;\n req
uired binary name (UTF8);\n}\n  }\n}\nrequired group params_complex (MAP) {\nrepeated group key_value {\nrequ
ired binary key (UTF8);\nrequired group value {\n  required int64 id;\n required binary name (UTF8);\n  }\n}\
n}\nrequired group params_struct_complex {\n  required int64 id;\n required group detail {\n  required int64
id;\n required binary name (UTF8);\n  }\n  }\n}",
        "dataxParquetMode": "fields"
      }
    }
```

5.3.14. PostgreSQL Writer

This topic describes the data types and parameters that are supported by PostgreSQL Writer and how to configure PostgreSQL Writer by using the codeless user interface (UI) and code editor.

PostgreSQL Writer writes data to tables stored in an PostgreSQL database. PostgreSQL Writer connects to a remote PostgreSQL database by using Java Database Connectivity (JDBC), and executes an SQL statement to write data to the PostgreSQL database.

Note Before you configure PostgreSQL Writer, you must configure a PostgreSQL data source. For more information, see Add a PostgreSQL data source.

- PostgreSQL Writer generates the SQL statement based on the table, column, and where parameters that you specified, and sends the generated SQL statement to the PostgreSQL database.
- If you specify the querySql parameter, PostgreSQL Writer directly sends the value of this parameter to the PostgreSQL database.

Precautions

If the name of a PostgreSQL table or a field starts with a digit or the name contains a letter or a hyphen (-), use double quotation marks (") to escape the name. If you do not escape the name, an error occurs when you use PostgreSQL Writer to write data to the PostgreSQL database. For PostgreSQL Writer, double quotation marks (") are keywords in JSON code. Therefore, you must use backslashes (\) to escape the double quotation marks (") that you use. For example, if the name of a PostgreSQL table is `123Test`, the name becomes `\"123Test\"` after it is escaped.

Note

- Each double quotation mark (") must be escaped by using a backslash (\).
- You can use only the code editor to perform escaping.

The following code provides an example on how to use the code editor to perform escaping:

```
"parameter": {
    "datasource": "abc",
    "column": [
        "id",
        "\"123Test\"", // Add escape characters.
],
"where": "",
"splitPk": "id",
"table": "public.wpw_test"
},
```

Data types

PostgreSQL Writer supports most PostgreSQL data types. Make sure that the data types of your database are supported.

The following table lists the data types supported by PostgreSQL Writer.

Data Integration data type	PostgreSQL data type
LONG	BIGINT, BIGSERIAL, INTEGER, SMALLINT, and SERIAL
DOUBLE	DOUBLE, PRECISION, MONEY, NUMERIC, and REAL
STRING	VARCHAR, CHAR, TEXT, BIT, and INET
DATE	DATE, TIME, and TIMESTAMP
BOOLEAN	BOOL
BYTES	BYTEA

Note

- PostgreSQL Writer supports only the data types that are listed in the preceding table.
- You can convert the MONEY, INET, and BIT data types by using syntax such as `a_inet::varchar`.

Parameters

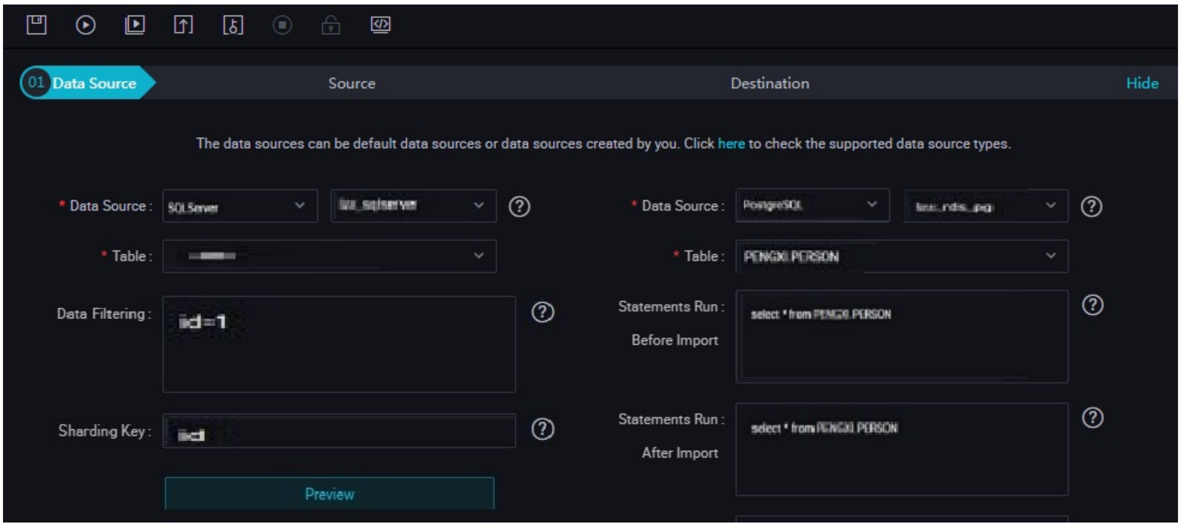
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table to which you want to write data.	Yes	No default value
writeMode	The write mode. Valid values: - *insert*: executes the `insert into...values...` statement to write data to the PostgreSQL database. If a primary key conflict or unique index conflict occurs, data cannot be written to the conflicting rows, and the data that is not written to these rows is regarded as dirty data. We recommend that you use the insert mode. - *copy*: copies data between tables and the standard input or output file. Data Integration supports the `COPY FROM` command, which allows you to copy data from files to tables. We recommend that you use this mode if performance issues occur.	No	*insert*
column	The names of the columns to which you want to write data. Separate the names with commas (,), `"column":["id","name","age"]`. If you want to write data to all the columns in the destination table, set this parameter to an asterisk (*), such as `"column":["*"]`.	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to delete outdated data. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to add a timestamp. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and PostgreSQL and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

<table>
<tr><th>Parameter</th><th>Description</th><th>Required</th><th>Default value</th></tr>
<tr><td>pgType</td><td>The PostgreSQL configuration for converting data types. Valid values: bigint[], double[], text[], jsonb, and json. Example:
<pre>
{
    "job":
    {
        "content":
        [{
            "reader": {...},
            "writer":
            {
                "parameter":
                {
                    "column":
                    [
                        // The names of the columns to which you want to write data.
                        "bigint_arr",
                        "double_arr",
                        "text_arr",
                        "jsonb_obj",
                        "json_obj"
                    ],
                    "pgType":
                    {
                        // The configuration that is used to convert data types. In each key-value pair, the key specifies the name of a field in the destination table, and the value specifies the data type of the field.
                        "bigint_arr": "bigint[]",
                        "double_arr": "double[]",
                        "text_arr": "text[]",
                        "jsonb_obj": "jsonb",
                        "json_obj": "json"
                    }
                }
            }
        }]
    }
}
</pre></td><td>No</td><td>No default value</td></tr>
</table>

Configure PostgreSQL Writer by using the codeless UI

1. Configure data sources.

Configure **Source** and **Target** for the synchronization node.



The data sources can be default data sources or data sources created by you. Click here to check the supported data source types.

Source:

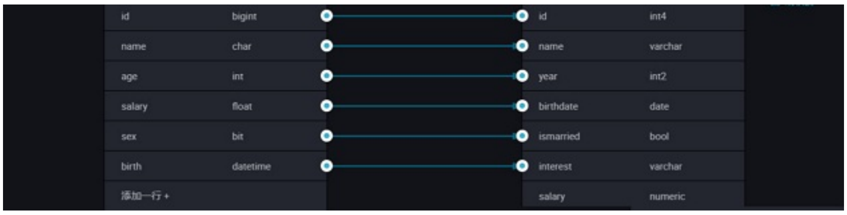
- * Data Source : SQLServer [redacted]
- * Table : [redacted]
- Data Filtering : id=1
- Sharding Key : id

Destination:

- * Data Source : PostgreSQL [redacted]
- * Table : PENGXI.PERSON
- Statements Run Before Import : select * from PENGXI.PERSON
- Statements Run After Import : select * from PENGXI.PERSON

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The table to which you want to write data. This parameter is equivalent to the table parameter that is described in the preceding section.
Statement Run Before Writing	The SQL statement that you want to execute before the synchronization node is run. This parameter is equivalent to the preSql parameter that is described in the preceding section.
Statement Run After Writing	The SQL statement that you want to execute after the synchronization node is run. This parameter is equivalent to the postSql parameter that is described in the preceding section.
Write Method	The write mode. This parameter is equivalent to the writeMode parameter that is described in the preceding section. Valid values: *insert* and *copy*.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.



id	bigint	id	int4
name	char	name	varchar
age	int	year	int2
salary	float	birthdate	date
sex	bit	ismarried	bool
birth	datetime	interest	varchar
添加一行 +		salary	numeric

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3.

Configure PostgreSQL Writer by using the code editor

You can configure PostgreSQL Writer by using the code editor. For more information, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to write data to a PostgreSQL database. For more information about the parameters, see the preceding parameter description.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"postgresql",// The writer type.
            "parameter":{
                "postSql":[],// The SQL statement that you want to execute after the synchronization node is ru
n.
                "datasource":"// The name of the data source.
                    "col1",
                    "col2"
                ],
                "table":"",// The name of the table to which you want to write data.
                "preSql":[]// The SQL statement that you want to execute before the synchronization node is run
.
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.15. Redis Writer

Redis Writer is a writer that is developed based on the Data Integration framework. It can write the data that is obtained from a reader to Redis.

REmote DIctionary Server (Redis) is a key-value storage system that can be accessed over the network and can permanently store data or store data in memory. Redis supports logs and provides high performance. You can use Redis as a database, a cache, or a message broker. Redis supports values of the following data types: STRING, LIST, SET, ZSET (sorted set), and HASH. For more information about Redis, visit redis.io.

Redis Writer interacts with a Redis server by using Jedis. Jedis is a Java client development kit that is provided and recommended by Redis.

Note

- Redis Writer can write data only to Redis clusters. The synchronization nodes that use Redis Writer must run on **exclusive resource groups for Data Integration**.
- Before you configure Redis Writer, you must add a Redis data source. For more information, see Add a Redis data source.
- If you rerun a synchronization node that uses Redis Writer to write LIST values to Redis, the result is not idempotent. In this case, you must manually delete the data written by Redis Writer in the previous run from Redis before you rerun the synchronization node.

Parameters

Parameter	Description	Required	Default value
expireTime	The expiration time of the values that are cached in Redis. Unit: seconds. If you do not configure this parameter, the default value `0` is used, which indicates that the values are permanently valid. You can specify the value of the expireTime parameter in one of the following modes: - **seconds**: the relative period of time in seconds starting from the current point in time. It specifies the period during which the values are valid. - **unixtime**: the number of seconds that have elapsed since 00:00:00 on January 1, 1970. It specifies the point in time when the values expire. Note If you set the expireTime parameter to a value that is greater than 2,592,000 (30 days), the value is interpreted as a **UNIX** timestamp.	No	*0*
keyFieldDelimiter	The delimiter that you want to use to separate keys when Redis Writer writes data to Redis. Example: key=key1\u0001id. If multiple keys need to be concatenated, this parameter is required. If the data contains only one key, this parameter is not required.	No	*\u0001*
dateFormat	The format in which Redis Writer writes the data of the DATE type to Redis. Set the value to yyyy-MM-dd HH:mm:ss.	No	No default value
datasource	The name of the data source. The name must be the same as that of the data source that you added.	Yes	No default value

Parameter	Description	Required	Default value
writeMode	The mode in which Redis Writer writes data to Redis. Redis Writer can write values of the following types: - STRING - LIST - SET - ZSET (sorted set) - HASH The value of the writeMode parameter varies based on the data types of the values that you want to write to Redis. For more information, see Configure writeMode. ⓘ **Note** When you configure Redis Writer, you can specify only one data type for the writeMode parameter. If you do not configure this parameter, the default value `string` is used.	No	*string*
keyIndexes	The indexes of source columns that are used as the key. The indexes start from 0. This indicates that the index of the first column is 0, and the index of the second column is 1, and so on. - If you want to specify a specific column of the source as the key, set this parameter to the index of the column. For example, if you want to specify the first column as the key, set this parameter to `0` . - If you want to specify a range of contiguous columns of the source as the key, set this parameter to a closed interval that specifies the indexes of these columns. For example, if you want to specify the second, third, and fourth columns as the key, set this parameter to `[1,3]` . ⓘ **Note** After you specify the keyIndexes parameter, Redis Writer specifies the remaining columns as the value. If you want to synchronize only some of the columns in the source, specify the names of the columns when you configure a reader.	Yes	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and Redis and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1,000*
timeout	The timeout period of the connection to Redis when data is written to Redis. Unit: milliseconds.	No	*30,000*
redisMode	The deployment mode of Redis. Redis Writer can write data only to Redis clusters. You must set this parameter to ClusterMode. ⓘ **Note** Synchronization nodes that use Redis Writer must run on exclusive resource groups for Data Integration.	No	No default value

Parameter	Description	Required	Default value
column	The names of the columns to which you want to write data. If you set the type parameter to string and the mode parameter to set, take note of the following points: • If the column parameter is not configured, the values stored in Redis are strings that are connected by delimiters. For example, in a CSV source file, the value of id is 1, the value of name is Bob, the value of age is 18, and the value of sex is male. In this case, the values to be written to Redis are "18::male". • If the column parameter is configured in the "column": [{"index":"0", "name":"id"}, {"index":"1", "name":"content"}] format, the values are stored in Redis in JSON format {"id":"Value of a source column","name":"Value of a source column"}. For example, the value of id is 1, and the value of name is Bob. In this case, the values stored in Redis are {"id":1,"name":"Bob"}.	No	No default value

Configure writeMode

When you configure Redis Writer, you can specify only one data type for the writeMode parameter. If you do not configure this parameter, the default value `string` is used.

Data type of values	type parameter (required)	mode parameter (required)	valueFieldDelimiter parameter (optional)	Configuration example

STRING	Set the type parameter to `string`.	The mode in which Redis Writer writes the data of the STRING type to Redis. Take note of the following rules when you set the type parameter to string:
		• Set the mode parameter to `set`.
		• If the data that you want to write to Redis already exists in Redis, existing data is overwritten.
			The delimiter that you want to use to separate column values. Default value: `\u0001`.
			• This parameter is required if you specify two or more columns as the value. For example, you specify three columns of the source data as the value and use the default delimiter. In this case, the column values are concatenated in the value1\u0001value2\u0001value3
				`"writeMode":{`
				`        "type": "string",`
				`        "mode": "set",`
				`        "valueFieldDelimiter":`
				`"\u0001"`
				`        }`

LIST	Set the type parameter to `list`.	The mode in which Redis Writer writes the data of the LIST type to Redis. Valid values:
		• `lpush`: indicates that data is stored at the leftmost of the list.
		• `rpush`: indicates that data is stored at the rightmost of the list.
				`"writeMode":{`
				`    "type": "list",`
				`    "mode": "lpush|rpush",`
				`    "valueFieldDelimiter":`
				`"\u0001"`
				`}`

Data type of values	type parameter (required)	mode parameter (required)	valueFieldDelimiter parameter (optional)	Configuration example

SET	Set the type parameter to `set` .	The mode in which Redis Writer writes the data of the SET type to Redis. Take note of the following rules when you set the type parameter to set:
		• Set the mode parameter to `sadd` , which indicates that data is stored to a set.
		• If the data that you want to write to Redis already exists in Redis, existing data is overwritten.
			format.
			• This parameter is not required if you specify only one column as the value.
				`"writeMode":{`
				`        "type": "set",`
				`        "mode": "sadd",`
				`        "valueFieldDelimiter":`
				`"\u0001"`
				`        }`

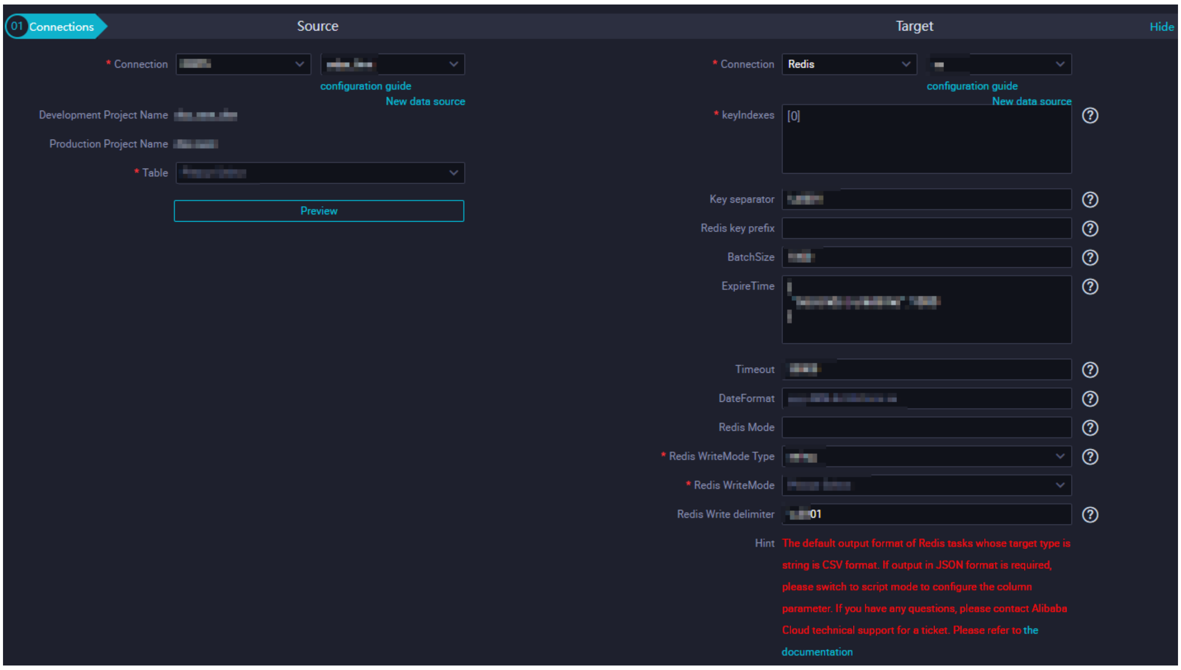
ZSET (sorted set)	Set the type parameter to `zset` .	The mode in which Redis Writer writes the data of the ZSET type to Redis. Take note of the following rules when you set the type parameter to zset:
		• Set the mode parameter to `zadd` , which indicates that data is stored to a sorted set.
		• If the data that you want to write to Redis already exists in Redis, existing data is overwritten.
			You do not need to configure this parameter.	`"writeMode":{`
				`        "type": "zset",`
				`        "mode": "zadd"`
				`        }`
				Note If the data type is set to ZSET, each row of the source data must meet the following requirements: A row can contain only one score and one value except for the key. The score must be placed before the value. This way, Redis Writer can distinguish between the score and value.

HASH	Set the type parameter to `hash` .	The mode in which Redis Writer writes the data of the HASH type to Redis. Take note of the following rules when you set the type parameter to hash:
		• Set the mode parameter to `hset` , which indicates that data is stored to a hash sorted set.
		• If the data that you want to write to Redis already exists in Redis, existing data is overwritten.
			You do not need to configure this parameter.	`"writeMode":{`
				`        "type": "hash",`
				`        "mode": "hset"`
				`        }`
				Note If the data type is set to HASH, each row of the source data must meet the following requirements: A row can contain only one attribute and one value except for the key. The attribute must be placed before the value. This way, Redis Writer can distinguish between the attribute and value.

Configure Redis Writer by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



Source

* Connection [redacted] [redacted]

Development Project Name [redacted]

Production Project Name [redacted]

* Table [redacted]

Target

* Connection Redis [redacted]

* keyIndexes [0]

Key separator [redacted]

Redis key prefix

BatchSize [redacted]

ExpireTime [redacted]

Timeout [redacted]

DateFormat [redacted]

Redis Mode

* Redis WriteMode Type [redacted]

* Redis WriteMode [redacted]

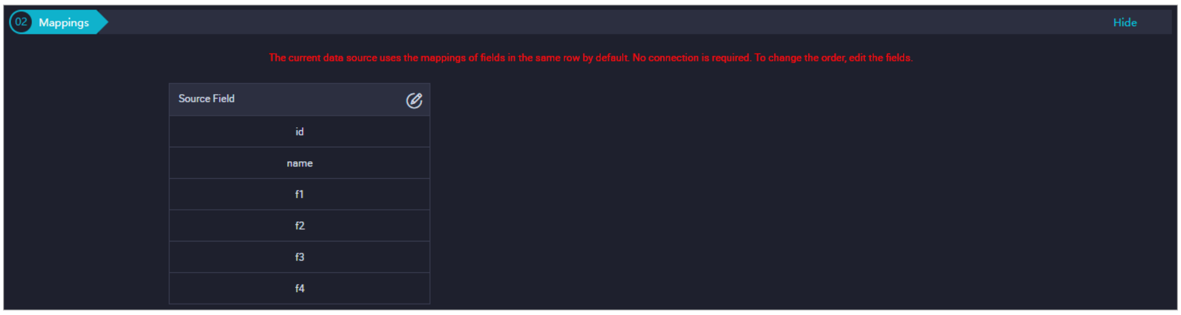
Redis Write delimiter [redacted]01

Hint The default output format of Redis tasks whose target type is string is CSV format. If output in JSON format is required, please switch to script mode to configure the column parameter. If you have any questions, please contact Alibaba Cloud technical support for a ticket. Please refer to the documentation

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the Parameters section.
keyIndexes	This parameter is equivalent to the keyIndexes parameter that is described in the Parameters section. This parameter specifies the indexes of source columns that are used as the key. The indexes start from 0. This indicates that the index of the first column is 0, and the index of the second column is 1, and so on.
Key separator	This parameter is equivalent to the keyFieldDelimiter parameter that is described in the Parameters section. This parameter specifies the delimiter that you want to use to separate keys when Redis Writer writes data to Redis.
Redis key prefix	The prefix for the key. For example, the prefix is `prefix::`, and the key is `1::2`. In this case, the key that is generated is `prefix::1::2`.
BatchSize	This parameter is equivalent to the batchSize parameter that is described in the Parameters section.
ExpireTime	This parameter is equivalent to the expireTime parameter that is described in the Parameters section.
Timeout	This parameter is equivalent to the timeout parameter that is described in the Parameters section.
DateFormat	This parameter is equivalent to the dateFormat parameter that is described in the Parameters section.
Redis Mode	This parameter is equivalent to the redisMode parameter that is described in the Parameters section.
Redis WriteMode Type	This parameter is equivalent to the writeMode parameter that is described in the Parameters section.
Redis WriteMode	The mode in which Redis Writer writes data to Redis. Valid values: set, lpush, rpush, sadd, zadd, and hset. This parameter is equivalent to the writeMode parameter that is described in the Parameters section. For more information, see Configure writeMode.

Parameter	Description
Redis Write delimiter	This parameter is equivalent to the **keyFieldDelimiter** parameter that is described in the Parameters section.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the Parameters section. The system maps the field in a row of the source to the field in the same row of the destination. You can click the icon to edit fields in the source.



The current data source uses the mappings of fields in the same row by default. No connection is required. To change the order, edit the fields.

3.

Configure Redis Writer by using the code editor

In the following code, a synchronization node is configured to write data from a MySQL database to Redis. For more information about the parameters, see Parameters.

> ⓘ **Note** For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0", // The version number.
    "steps":[
        { // The following sample code is used to configure the reader. For more information about the paramete
rs of the reader, see the topic of the related reader.
            "stepType":"mysql",
            "parameter": {
                "envType": 0,
                "datasource": "xc_mysql_demo2",
                "column": [
                    "id",
                    "value",
                    "table"
                ],
                "connection": [
                    {
                        "datasource": "xc_mysql_demo2",
                        "table": []
                    }
                ],
                "where": "",
                "splitPk": "",
                "encoding": "UTF-8"
            },,
            "name":"Reader",
            "category":"reader"
        },
        {// The following sample code is used to configure the writer:
            "stepType":"redis",                    // The writer type. Set the value to redis.
            "parameter":{                          // Configure the following parameters of Redis Writer. For m
ore information about the parameters, see Parameters.
                "expireTime":{                     // The expiration time of the values that are cached in Redi
s. Set the value to a period of time in seconds or a UNIX timestamp. For example, you can set this parameter to
"seconds":"1000".
                            },
                "keyFieldDelimiter":"u0001",       // The delimiter that you want to use to separate keys when
Redis Writer writes data to Redis.
                "dateFormat":"yyyy-MM-dd HH:mm:ss",// The format in which Redis Writer writes the data of the D
ATE type to Redis.
                "datasource":"xc_mysql_demo2",     // The name of the data source. The name must be the same as
that of the data source that you added.
                "envType": 0,                      // The type of the environment. Set the value to 1 for the d
evelopment environment. Set the value to 0 for the production environment.
                "writeMode":{                      // The mode in which data is written to Redis.
                    "type": "string",                 // The data type of the value that you want to write to R
edis.
                    "mode":"set",                  // The mode in which the data of a type specified by the typ
e parameter is written to Redis.
                    "valueFieldDelimiter":"u0001", // The delimiter that you want to use to separate column val
ues.
                             },
                "keyIndexes":[0,1],                // The indexes of source columns that are used as the key wh
en Redis Writer writes data to Redis. In this example, the value is set to [0,1], which indicates that the firs
t and second columns of the source data are used as the key.
                "batchSize":"1000"                 // The number of data records to write at a time.
        "column": [                        // If you set the type parameter to string and the mode parameter to
set and do not configure the column parameter, the values to be written to Redis are strings that are connected
by delimiters. For example, for a CSV source file, if the value of id is 1, the value of name is Bob, the value
of age is 18, and the value of sex is male, the values to be written to Redis are "18::male". If you set the ty
pe parameter to string and the mode parameter to set and configure the column parameter in the following format
, the values are written to Redis in JSON format, such as {"id":1,"name":"Bob","age":18,"sex":"male"}.
                {
                "name": "id",
                "index": "0"
                },
                {
                "name": "name",
                "index": "1"
                },
                {
                "name": "age",
                "index": "2"
                },
                {
                "name": "sex",
                "index": "3"
                }
            ]
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"                           // The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.16. SQL Server Writer

This topic describes the data types and parameters that are supported by SQL Server Writer and how to configure SQL Server Writer by using the codeless user interface (UI) and code editor.

SQL Server Writer writes data to tables stored in primary SQL Server databases. SQL Server Writer connects to a remote SQL Server database by using Java Database Connectivity (JDBC) and executes the `INSERT INTO` statement to write data to the SQL Server database. Data is written to the database in batches.

Note Before you configure SQL Server Writer, you must configure an SQL Server data source. For more information, see Configure an SQL Server data source.

SQL Server Writer is designed for extract, transform, load (ETL) developers to import data in data warehouses to SQL Server databases. SQL Server Writer can also be used as a data migration tool by users such as database administrators.

SQL Server Writer obtains data from a reader and writes the data to the destination database by executing the `INSERT INTO` statement. If a primary key conflict or unique index conflict occurs, data cannot be written to the conflicting rows. To improve performance, SQL Server Writer performs batch updates by using a `PreparedStatement object` and sets the `rewriteBatchedStatements` parameter to true. This way, SQL Server Writer buffers data and submits a write request when the volume of data in the buffer reaches a specified threshold.

Note

- Data can be written only to tables that are stored in primary SQL Server databases.
- A synchronization node that uses SQL Server Writer must have at least the permissions to execute the `INSERT INTO` statement. Whether other permissions are required depends on the SQL statements that you specify in the preSql and postSql parameters when you configure the node.

SQL Server versions

SQL Server Writer uses the driver com.microsoft.sqlserver sqljdbc4 4.0. For more information about the capabilities of the driver, see the official documentation. The following table lists the commonly used SQL Server versions and describes whether they are supported by the driver.

Version	Supported
SQL Server 2016	Yes
SQL Server 2014	Yes
SQL Server 2012	Yes
PDW 2008R2 AU34	Yes
SQL Server 2008 R2	Yes
SQL Server 2008	Yes
SQL Server 2019	No
SQL Server 2018	No

Data types

SQL Server Writer supports most SQL Server data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by SQL Server Writer.

Category	SQL Server data type
Integer	BIGINT, INT, SMALLINT, and TINYINT
Floating point	FLOAT, DECIMAL, REAL, and NUMERIC
String	CHAR, NCHAR, NTEXT, NVARCHAR, TEXT, VARCHAR, NVARCHAR (MAX), and VARCHAR (MAX)
Date and time	DATE, TIME, and DATETIME
Boolean	BIT
Binary	BINARY, VARBINARY, VARBINARY (MAX), and TIMESTAMP

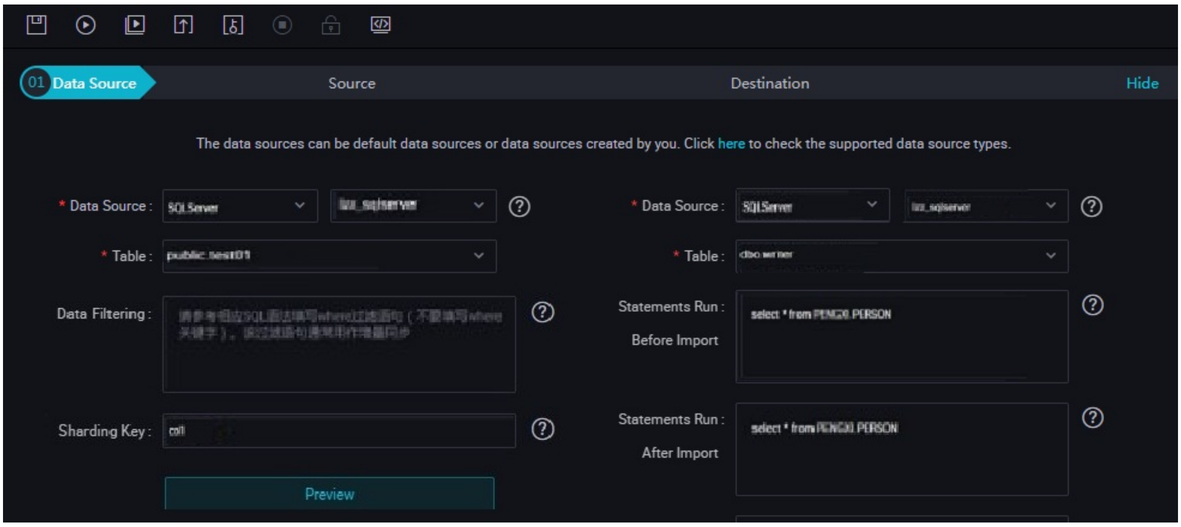
Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table to which you want to write data.	Yes	No default value
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as `"column":["id","name","age"]` . If you want to write data to all the columns in the destination table, set this parameter to an asterisk (*), such as `"column":["*"]` .	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to delete outdated data. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to add a timestamp. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value
writeMode	The write mode. Valid value: *insert*. If a primary key conflict or unique index conflict occurs, Data Integration considers data as dirty data and retains the original data.	No	*insert*
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and SQL Server and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

Configure SQL Server Writer by using the codeless UI

1. Configure data sources.

Configure Source and **Target** for the synchronization node.



The data sources can be default data sources or data sources created by you. Click here to check the supported data source types.

Source:

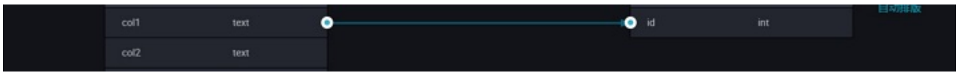
- * Data Source : SQLServer | lzz_sqlserver
- * Table : public.test01
- Data Filtering : 请参考相应SQL语法填写where过滤语句（不要填写where关键字）。该过滤语句通常用作增量同步
- Sharding Key : col1

Destination:

- * Data Source : SQLServer | lzz_sqlserver
- * Table : dbo.writer
- Statements Run Before Import : select * from PENGXI.PERSON
- Statements Run After Import : select * from PENGXI.PERSON

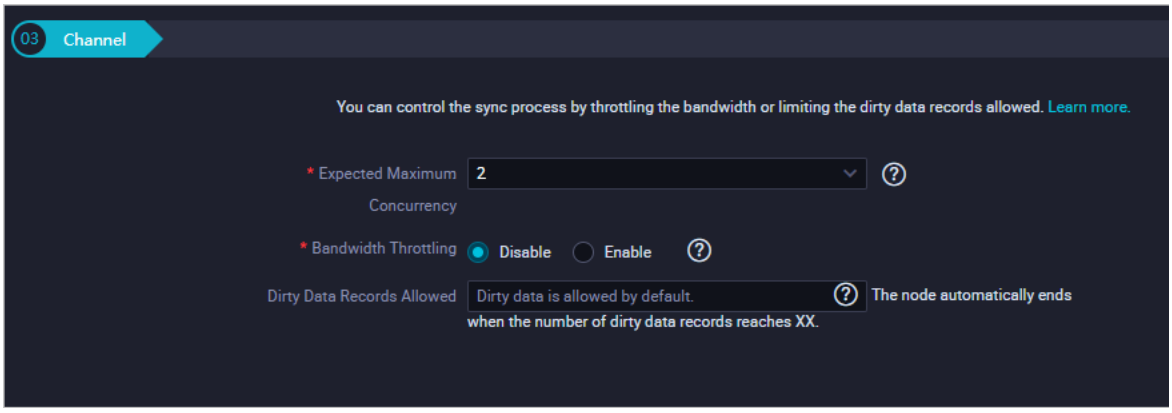
Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table to which you want to write data. This parameter is equivalent to the table parameter that is described in the preceding section.
Statement Run Before Writing	The SQL statement that you want to execute before the synchronization node is run. This parameter is equivalent to the preSql parameter that is described in the preceding section.
Statement Run After Writing	The SQL statement that you want to execute after the synchronization node is run. This parameter is equivalent to the postSql parameter that is described in the preceding section.
Solution to Duplicate Primary Keys	The write mode. This parameter is equivalent to the writeMode parameter that is described in the preceding section.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.



Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: Dirty data is allowed by default. The node automatically ends when the number of dirty data records reaches XX.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.

4.

Configure SQL Server Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to write data to an SQL Server database. For more information about the parameters, see the preceding parameter description.

Note Delete the comments from the following code before you run the code.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"sqlserver",// The writer type.
            "parameter":{
                "postSql":[],// The SQL statement that you want to execute after the synchronization node is ru
n.
                "datasource":"",// The name of the data source.
                "column":[// The names of the columns to which you want to write data.
                    "id",
                    "name"
                ],
                "table":"",// The name of the table to which you want to write data.
                "preSql":[]// The SQL statement that you want to execute before the synchronization node is run
.
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.17. Lindorm Writer

This topic describes the data types and parameters that are supported by Lindorm Writer and how to configure Lindorm Writer by using the codeless user interface (UI) and code editor.

Context

Lindorm Writer writes data to tables stored in ApsaraDB for Lindorm databases. Lindorm Writer connects to a remote ApsaraDB for Lindorm database by using a Java client and calls API operations to write data to tables of the table and wideColumn types stored in the ApsaraDB for Lindorm database.

Note

- The configuration parameter is required for Lindorm Writer. You can go to the ApsaraDB for Lindorm console to obtain the configuration items that are necessary for Data Integration to connect to an ApsaraDB for Lindorm cluster. The configuration data must be in the JSON format.
- ApsaraDB for Lindorm is a multimode database. Lindorm Writer writes data to the model tables of the table and wideColumn types stored in ApsaraDB for Lindorm databases. For more information about the model tables of the table and wideColumn types, see Overview. You can also consult Lindorm engineers on duty by using DingTalk.

Data types

Lindorm Writer supports most ApsaraDB for Lindorm data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by Lindorm Writer.

Category	ApsaraDB for Lindorm data type
Integer	INT, LONG, and SHORT
Floating point	DOUBLE, FLOAT, and DOUBLE
String	STRING
Date and time	DATE
Boolean	BOOLEAN
Binary	BINARYSTRING

Parameters

Parameter	Description	Required	Default value
configuration	The configuration items that are necessary for Data Integration to connect to each ApsaraDB for Lindorm cluster. You can go to the ApsaraDB for Lindorm console to obtain the configuration items and ask the administrator of the ApsaraDB for Lindorm database to convert the configurations to data in the following JSON format: *{"key1":"value1","key2":"value2"}*. Example: *{"lindorm.zookeeper.quorum":"???? "," lindorm.zookeeper.property.clientPort ":"???? "}*. Note If you write the JSON code manually, you must escape double quotation marks (") by using \".	Yes	No default value
dynamicColumn	The dynamic column mode. The configurations of this mode are complex and this mode is not used in most cases. Valid values: true and false. Default value: false.	Yes	false
table	The name of the table to which you want to write data. The table name is case-sensitive.	Yes	No default value
namespace	The namespace of the table to which you want to write data. The namespace of the table is case-sensitive.	Yes	No default value
encoding	The encoding method. Valid values: UTF-8 and GBK. This parameter is used to convert the lindorm byte[] data stored in binary mode to strings.	No	UTF-8

Parameter	Description	Required	Default value
columns	The columns of the table to which you want to write data. Lindorm Writer allows you to write data to the specified columns in a destination table in an order different from that specified in the schema of the source table. - If Lindorm Writer writes data to a table of the table type stored in ApsaraDB for Lindorm databases, you need only to specify the column names in the table. The column type information is automatically obtained based on the metadata of the source table and is filled in the destination table. - Lindorm Writer writes data to a table of the table or wideColumn type stored in ApsaraDB for Lindorm databases.	Yes	No default value

Configure Lindorm Writer by using the codeless UI

This method is not supported.

Configure Lindorm Writer by using the code editor

- For more information about how to configure a job that writes data from a MySQL data source to a table of the table type in an ApsaraDB for Lindorm database by using the code editor, see Create a synchronization node by using the code editor.

  Note Delete the comments from the following code before you run the code:

```
{
    "type": "job",
    "version": "2.0",
    "steps": [
        {
            "stepType": "mysqlreader",
            "parameter": {
                "checkSlave": true,
                "datasource": "_IDB.TAOBAO",
                "envType": 1,
                "column": [
                    "id",
                    "value",
                    "table"
                ],
                "socketTimeout": 3600000,
                "masterSlave": "slave",
                "connection": [
                    {
                        "datasource": "_IDB.TAOBAO",
                        "table": []
                    }
                ],
                "where": "",
                "splitPk": "",
                "encoding": "UTF-8",
            "print": true
            },
            "name": "mysqlreader",
            "category": "reader"
        },
        {
            "stepType": "lindormwriter",
            "parameter": {
                "configuration": "The configuration items that are necessary for Data Integration to connect
to each ApsaraDB for Lindorm cluster. The value is in the JSON format.",
to each ApsaraDB for Lindorm cluster. The value is in the JSON format.",
                "nullMode": "skip",
                "datasource": "_LINDORM",
                "envType": 1,
                "columns": "columns": [
                    "id",
                    "name",
                    "age",
                    "birthday",
                    "gender"
                ],
                "guid": "",
                "hbaseColumn": "",
                "dynamicColumn": "false",
                "table": "lindorm_table",
                "encoding": "utf8",
                "hbaseRowkey": ""
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "setting": {
        "jvmOption": "",
        "executeMode": null,
        "errorLimit": {
            "record": "0"
        },
       "speed": {
        // The transmission rate, in Byte/s. Data Integration runs to reach this rate as much as possible but
does not exceed it.
        "byte": 1048576
      }
       // The maximum number of dirty data records allowed.
       "errorLimit": {
        // The maximum number of dirty data records allowed. If the value of errorlimit is greater than the m
aximum value, an error is reported.
        "record": 0,
        // The maximum percentage of dirty data records. 1.0 indicates 100% and 0.02 indicates 2%.
        "percentage": 0.02
      }
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}
```

- For more information about how to configure the job that writes data from a MySQL data source to a table of the wideColumn type stored in an ApsaraDB for Lindorm database by using the code editor, see Create a synchronization node by using the code editor.

> Note Delete the comments from the following code before you run the code:

```
{
  "job": {
    "setting": {
      "speed": {
        // The transmission rate, in Byte/s. Data Integration runs to reach this rate as much as possible but
does not exceed it.
        "byte": 1048576
      }
      // The maximum number of dirty data records allowed.
      "errorLimit": {
        // The maximum number of dirty data records allowed. If the value of errorlimit is greater than the m
aximum value, an error is reported.
        "record": 0,
        // The maximum percentage of dirty data records. 1.0 indicates 100% and 0.02 indicates 2%.
        "percentage": 0.02
      }
    },
    "content": [
      {
        "reader": {
          // The reader type.
          "name": "mysqlreader",
          // Specifies whether to print data.
          "parameter": {
            "print": true
          }
    },
        "writer": {
          "name": "lindormriter",
          "parameter": {
            "configuration": "The configuration items that are necessary for Data Integration to connect to e
ach ApsaraDB for Lindorm cluster. The value is in the JSON format.",
            "namespace": "namespace",
             "table": "lindorm_table",
            "encoding": "utf8",
            "nullMode": "skip",
            "dynamicColumn": "false",
            "caching": 128,
            "dynamicColumn": "false",
            "columns": "columns": [
                    "ROW|STRING",
                    "cf:id|STRING",
                    "cf:age|INT",
                    "cf:birthday|STRING"
                ]
           }
        }
      }
    ]
  }
}
```

5.3.18. Elasticsearch Writer

This topic describes the parameters that are supported by Elasticsearch Writer and how to configure Elasticsearch Writer by using the code editor.

Limits

You can add Elasticsearch V5.X, V6.X, and V7.X data sources to DataWorks. Self-managed Elasticsearch data sources are not supported.

Background information

Elasticsearch Writer can write data to Elasticsearch V5.X clusters by using the shared resource group for Data Integration and to Elasticsearch V5.X, V6.X, and V7.X clusters by using exclusive resource groups for Data Integration. For more information about exclusive resource groups for Data Integration, see Create and use an exclusive resource group for Data Integration.

Elasticsearch is an open source product that is released under the Apache License. It is a popular search engine for enterprises. Elasticsearch is a distributed search and analytics engine built on top of Apache Lucene. The following description provides the mappings between the core concepts of Elasticsearch and those of a relational database:

```
Relational database instance  -> Database  -> Table -> Row       -> Column
Elasticsearch                 -> Index     -> Type  -> Document  -> Field
```

Elasticsearch can contain multiple indexes (databases). Each index can contain multiple types (tables). Each type can contain multiple documents (rows). Each document can contain multiple fields (columns). Elasticsearch Writer obtains data records from a reader and uses the RESTful API of Elasticsearch to write the data records to Elasticsearch in batches.

Parameters

Parameter	Description	Required	Default value
endpoint	The endpoint of Elasticsearch. Specify the endpoint in the `http://example.com:9999` format.	No	No default value
accessId	The AccessKey ID that is used to connect to the destination Elasticsearch cluster. The AccessKey ID is used for authentication before a connection to the Elasticsearch cluster can be established. **Note** The accessId and accessKey parameters are required. If you do not specify the parameters, an error is returned. If you use a self-managed Elasticsearch cluster for which basic access authentication is not configured, the AccessKey ID and AccessKey secret are not required. In this case, you can set the accessId and accessKey parameters to random values.	No	No default value
accessKey	The AccessKey secret that is used to connect to the destination Elasticsearch cluster.	No	No default value
index	The name of the index in the destination Elasticsearch cluster.	No	No default value
indexType	The name of the index type in the destination Elasticsearch cluster.	No	*Elasticsearch*
cleanup	Specifies whether to delete the existing data from the index. To delete the existing data, you must delete and recreate the index. The default value of this parameter is *false*, which indicates that the existing data in the index is retained.	No	*false*
batchSize	The number of data records to write at a time.	No	*1,000*
trySize	The maximum number of retries allowed after a failure.	No	*30*
timeout	The timeout period of the connection to the client.	No	*600,000*
discovery	Specifies whether to enable node discovery. If node discovery is enabled, the server list in the client is polled and regularly updated.	No	*false*
compression	Specifies whether to enable compression for an HTTP request.	No	*true*
multiThread	Specifies whether to use multiple threads for an HTTP request.	No	*true*
ignoreWriteError	Specifies whether to ignore write errors and proceed with data write without retries.	No	*false*
ignoreParseError	Specifies whether to ignore format parsing errors and proceed with data write.	No	*true*

Parameter	Description	Required	Default value
alias	The alias feature of Elasticsearch is similar to the view feature of a database. For example, if you create an alias named my_index_alias for the index my_index, the operations on my_index_alias also take effect on my_index. If you configure the alias parameter, the alias that you specify in this parameter is created for the index after data is written to the index.	No	No default value
aliasMode	The mode in which an alias is added after data is written to the index. Valid values: *append* and *exclusive*. • If you set the aliasMode parameter to *append*, an alias is added for the index. One alias maps multiple indexes. • If you set the aliasMode parameter to *exclusive*, the existing alias of the index is deleted and a new alias is added for the index. One alias maps one index. Elasticsearch Writer can convert aliases to actual index names. You can use aliases to migrate data from one index to another index, search for data across multiple indexes in a unified manner, and create a view on a subset of data in an index.	No	*append*
splitter	The delimiter (-,-) based on which Elasticsearch Writer splits the source data if the source data is an array. For example, the source column stores the `["a", "b", "c", "d"]` string. In this case, Elasticsearch Writer splits the data based on the delimiter (-,-), obtains the array `a-,-b-,-c-,-d` , and then writes the array to the related field in the destination Elasticsearch cluster.	No	-,-
settings	The settings of the index. The settings must follow official Elasticsearch specifications.	No	No default value
	The fields of the document. The parameters for each field include basic parameters such as name and type, and advanced parameters such as analyzer, format, and array. Elasticsearch supports the following field types:		

Parameter	Description	Required	Default value

column	`- id // The id type corresponds to the _id type in`
	`Elasticsearch and can be considered as the unique`
	`primary key. Data that has the same ID will be`
	`overwritten and not be indexed.`
	`- string`
	`- text`
	`- keyword`
	`- long`
	`- integer`
	`- short`
	`- byte`
	`- double`
	`- float`
	`- date`
	`- boolean`
	`- binary`
	`- integer_range`
	`- float_range`
	`- long_range`
	`- double_range`
	`- date_range`
	`- geo_point`
	`- geo_shape`
	`- ip`
	`- token_count`
	`- array`
	`- object`
	`- nested`
	The following information describes the field types:
	• If the field type is text, you can configure the analyzer, norms, and index_options parameters. Example:
	`{`
	`    "name": "col_text",`
	`    "type": "text",`
	`    "analyzer": "ik_max_word"`
	`    }`
	• If the field type is date, you can configure the format and timezone parameters to indicate the date serialization format and the time zone. You can also configure the origin parameter instead of the timezone parameter.
	○ If you configure the origin parameter, Elasticsearch Writer updates the mappings of the index and writes data to the index in the original format. We recommend that you configure the origin parameter.
	○ If you want to use Data Integration to convert the time zone, delete the origin parameter and configure the timezone parameter.
	Example:
	`{`
	`    "name": "col_date",`
	`    "type": "date",`
	`    "format": "yyyy-MM-dd HH:mm:ss",`
	`    "origin": true`
	`    }`
	• If the field type is geo_shape, you can configure the tree (geohash or quadtree) and precision parameters. Example:
		Yes	No default value

Parameter	Description	Required	Default value
dynamic	If you set this parameter to *true*, Elasticsearch Writer uses the mapping configuration of the destination Elasticsearch cluster instead of the mapping configuration of Data Integration. In Elasticsearch V7.X, the default value of the type parameter is _doc. If you use the mapping configuration of the destination Elasticsearch cluster, set the type parameter to _doc and the esVersion parameter to 7. You must add the following parameter configuration that specifies the version information to the code: `"esVersion": "7"` .	No	*false*
actionType	The type of action for writing data to the destination Elasticsearch cluster. Data Integration supports only the following action types: *index* and *update*. Default value: *index*. ● *index*: Elasticsearch Writer uses Index.Builder of an Elasticsearch SDK to construct a request for writing multiple data records at a time. In *index* mode, Elasticsearch Writer first checks whether an ID is specified for the document that you want to insert. ○ If no ID is specified, Elasticsearch Writer generates a unique ID. In this case, the document is directly inserted into the destination Elasticsearch cluster. ○ If an ID is specified, the existing document is replaced with the document that you want to insert. You cannot modify specific fields in the document. **Note** The replace operation in this case is different from that in Elasticsearch where specific fields can be modified. ● *update*: Elasticsearch Writer uses Update.Builder of an Elasticsearch SDK to construct a request for writing multiple data records at a time. In *update* mode, Elasticsearch Writer calls the get method of InternalEngine to obtain the information about the original document for each update. This way, you can modify specific fields. In update mode, you must obtain the information about the original document for each update, which greatly affects the performance. However, you can modify specific fields in this mode. If the original document does not exist, the new document is directly inserted.	No	*index*

Configure Elasticsearch Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to write data to Elasticsearch. For more information about the parameters, see the preceding parameter description.

```
{
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "steps": [
        {
            "category": "reader",
            "name": "Reader",
            "parameter": {
            },
            "stepType": "stream"
        },
        {
            "category": "writer",
            "name": "Writer",
            "parameter": {
                "endpoint": "http://example.com:9999",
                "accessId": "xxxx",
                "accessKey": "yyyy",
                "index": "test-1",
                "type": "default",
                "cleanup": true,
                "settings": {
                    "index": {
                        "number_of_shards": 1,
                        "number_of_replicas": 0
                    }
                },
                "discovery": false,
                "batchSize": 1000,
                "splitter": ",",
                "column": [
                    {
                        "name": "pk",
                        "type": "id"
                    },
                    {
                        "name": "col_ip",
                        "type": "ip"
                    },
                    {
                        "name": "col_double",
                        "type": "double"
                    },
                    {
                        "name": "col_long",
                        "type": "long"
                    },
                    {
                        "name": "col_integer",
                        "type": "integer"
                    },
                    {
                        "name": "col_keyword",
                        "type": "keyword"
                    },
                    {
                        "name": "col_text",
                        "type": "text",
                        "analyzer": "ik_max_word"
                    },
                    {
                        "name": "col_geo_point",
                        "type": "geo_point"
                    },
                    {
                        "name": "col_date",
                        "type": "date",
                        "format": "yyyy-MM-dd HH:mm:ss"
                    },
                    {
                        "name": "col_nested1",
                        "type": "nested"
                    },
                    {
                        "name": "col_nested2",
                        "type": "nested"
                    },
                    {
                        "name": "col_object1",
                        "type": "object"
                    },
                    {
                        "name": "col_object2",
                        "type": "object"
                    },
                    {
                        "name": "col_integer_array",
                        "type": "integer",
                        "array": true
                    },
                    {
                        "name": "col_geo_shape",
                        "type": "geo_shape",
                        "tree": "quadtree",
                        "precision": "10m"
                    }
                ]
            },
            "stepType": "elasticsearch"
        }
    ],
    "type": "job",
    "version": "2.0"
}
```

Note A connection failure may occur if you use the shared resource group for Data Integration to connect to an Elasticsearch cluster that is deployed in a virtual private cloud (VPC). To write data to an Elasticsearch cluster that is deployed in a VPC, use exclusive or custom resource groups for Data Integration. For more information about how to create an exclusive or custom resource group for Data Integration, see Exclusive resources for Data Integration or Create a custom resource group for Data Integration.

5.3.19. LogHub (SLS) Writer

This topic describes the data types and parameters that are supported by LogHub (SLS) Writer and how to configure LogHub (SLS) Writer by using the codeless user interface (UI) and code editor.

LogHub (SLS) Writer can transfer data from a reader to LogHub (SLS) by using Log Service SDK for Java.

Note LogHub (SLS) does not ensure idempotence. If you rerun a failed node, redundant data may be generated.

LogHub (SLS) Writer obtains data from a reader and converts the data types supported by Data Integration to STRING. When the number of the data records reaches the value specified for the batchSize parameter, LogHub (SLS) Writer sends the data records to LogHub at a time by using Log Service SDK for Java. The default value of the batchSize parameter is *1024*. The maximum value is *4096*.

Data types

The following table lists the data types that are supported by LogHub (SLS) Writer.

Data Integration data type	LogHub (SLS) data type
LONG	STRING
DOUBLE	STRING
STRING	STRING
DATE	STRING
BOOLEAN	STRING
BYTES	STRING

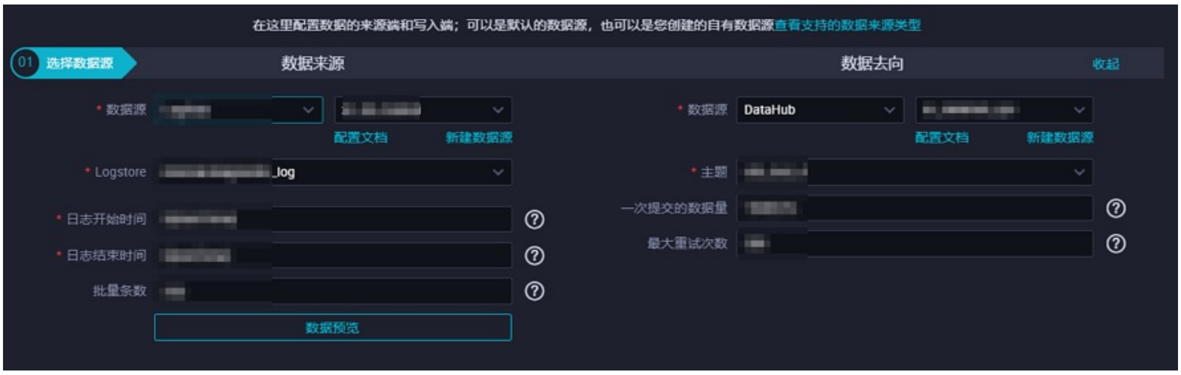
Parameters

Parameter	Description	Required	Default value
endpoint	The endpoint of the Log Service project. The endpoint is a URL that you can use to access the project and the log data in the project. The endpoint varies based on the project name and the Alibaba Cloud region where the project resides. For more information about the Log Service endpoints in each region, see Endpoints.	Yes	No default value
accessKeyId	The **AccessKey ID** of the Alibaba Cloud account that is used to access the Log Service project.	Yes	No default value
accessKeySecret	The AccessKey secret of the Alibaba Cloud account that is used to access the Log Service project.	Yes	No default value
project	The name of the Log Service project to which you want to write data.	Yes	No default value
logstore	The name of the Logstore to which you want to write data. A Logstore is a basic unit that you can use to collect, store, and query log data in Log Service.	Yes	No default value
topic	The name of the topic to which you want to write data.	No	Empty string
batchSize	The number of data records to write to LogHub (SLS) at a time. Default value: 1024. **Note** The size of the data to write to LogHub (SLS) at a time cannot exceed 5 MB. You can change the value of this parameter based on the size of a single data record.	No	*1024* (1,024 data records are written to LogHub at a time.)
column	The names of columns in each data record.	Yes	No default value

Configure LogHub (SLS) Writer by using the codeless UI

1. Configure data sources.

Configure **Source** and **Target** for the synchronization node.



在这里配置数据的来源端和写入端；可以是默认的数据源，也可以是您创建的自有数据源查看支持的数据来源类型

01 选择数据源 数据来源 数据去向 收起

数据来源:

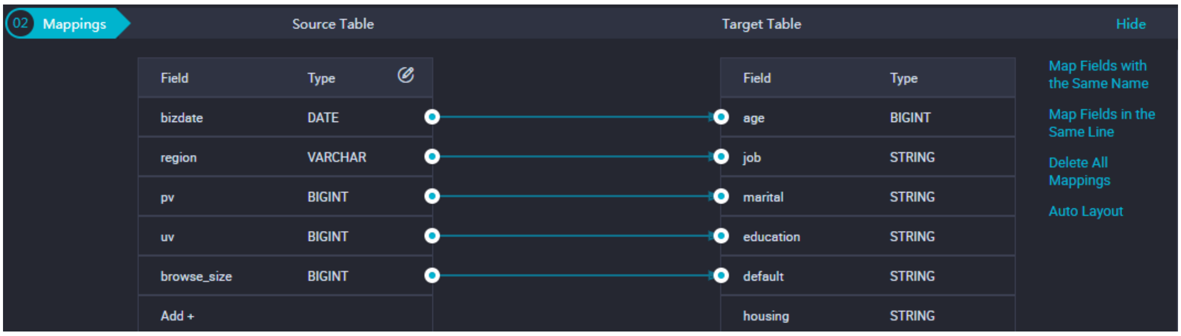
- * 数据源 [redacted] [redacted] 配置文档 新建数据源
- * Logstore [redacted]_log
- * 日志开始时间 [redacted]
- * 日志结束时间 [redacted]
- 批量条数 [redacted]
- 数据预览

数据去向:

- * 数据源 DataHub [redacted] 配置文档 新建数据源
- * 主题 [redacted]
- 一次提交的数据量 [redacted]
- 最大重试次数 [redacted]

Parameter	Description
Connection	The name of the data source to which you want to write data.
Logstore	The name of the Logstore to which you want to write data. This parameter is equivalent to the logstore parameter that is described in the preceding section.
Topic	The name of the topic to which you want to write data. This parameter is equivalent to the topic parameter that is described in the preceding section.
Number of batches	The number of data records to write to LogHub (SLS) at a time. This parameter is equivalent to the batchSize parameter that is described in the preceding section.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.



02 Mappings Source Table Target Table Hide

Field	Type	Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
Add +		housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3.

Configure LogHub (SLS) Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to write data to LogHub (SLS). For more information about the parameters, see the preceding parameter description.

```
{
    "type": "job",
    "version": "2.0",// The version number.
    "steps": [
        {
            "stepType": "stream",
            "parameter": {},
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "loghub",// The writer type.
            "parameter": {
                "datasource": "",// The name of the data source to which you want to write data.
                "column": [// The names of the columns to which you want to write data.
                    "col0",
                    "col1",
                    "col2",
                    "col3",
                    "col4",
                    "col5"
                ],
                "topic": "",// The name of the topic to which you want to write data.
                "batchSize": "1024",// The number of data records to write at a time.
                "logstore": ""// The name of the Logstore to which you want to write data.
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "setting": {
        "errorLimit": {
            "record": ""// The maximum number of dirty data records allowed.
        },
        "speed": {
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":3, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}
```

5.3.20. Open Search Writer

This topic describes how Open Search Writer works, its features, data types, and parameters, and how to configure it by using the code editor.

Notice Open Search Writer supports only exclusive resource groups for Data Integration, but not the default resource group or custom resource groups. For more information, see Create and use an exclusive resource group for Data Integration, Use the default resource group, and Add a custom resource group.

How it works

Open Search Writer allows you to insert data to or update data in Open Search. Open Search Writer is designed for developers to import data to Open Search so that the data can be searched.

Specifically, Open Search Writer uses the search API that is provided by Open Search to import data.

Note

- Open Search V3 uses internal dependent databases, with POM of com.aliyun.opensearch aliyun-sdk-opensearch 2.1.3.
- To use Open Search Writer, you must install JDK 1.6-32 or later. You can run the `java-version` command to view the JDK version.
- A sync node that is run on the default resource group may fail to connect to Open Search that is deployed in a virtual private cloud (VPC).

Features

The columns in Open Search are unordered. Open Search Writer writes data in strict accordance with the order of the specified columns. If the number of specified columns is less than that in Open Search, redundant columns in Open Search are set to the default value or null.

Assume that an Open Search table contains columns a, b, and c, and you only need to write data to columns b and c. You can set the column parameter to ["c","b"]. In this case, Open Search Writer imports the first and second columns of the source data that is obtained from a reader to columns c and b in the Open Search table. Column a in the Open Search table is set to the default value or null.

Additional instructions:

- Handling of column configuration errors

  To avoid losing the data of redundant columns and ensure high data reliability, Open Search Writer returns an error message if the number of columns to be written is more than that in the destination Open Search table. For example, if an Open Search table contains columns a, b, and c, Open Search Writer returns an error if more than three columns are to be written to the table.

- Table configuration

  Open Search Writer can write data to only one table at a time.

- Node rerunning

  After a node is rerun, data is overwritten based on IDs. Therefore, the data written to Open Search must contain an ID column. An ID is a unique identifier of a row in Open Search. The existing data with the same ID as the new data will be overwritten.

- Node rerunning

  After a node is rerun, data is overwritten based on IDs.

Data types

Open Search Writer supports most Open Search data types. Make sure that your data types are supported.

The following table describes the data types that Open Search Writer supports.

Category	Open Search data type
Integer	INT
Floating point	DOUBLE and FLOAT
String	TEXT, LITERAL, and SHORT_TEXT
Date and time	INT
Boolean	LITERAL

Parameters

Parameter	Description	Required	Default value
accessId	The AccessKey ID of the account that you can use to connect to the Open Search project.	Yes	N/A
accessKey	The AccessKey secret of the account that you can use to connect to the Open Search project.	Yes	N/A
host	The endpoint of Open Search. You can view the endpoint in the Alibaba Cloud Management Console.	Yes	N/A
indexName	The name of the Open Search project.	Yes	N/A
table	The name of the table to which data is written. You can specify only one table because Data Integration cannot import data to multiple tables at a time.	Yes	N/A
column	The columns in the destination table to which data is written. To write data to all the columns in the destination table, set the value to an asterisk (*), for example, `"column":["*"]` . Set the value to the specified columns if data needs to be written to only specific columns in the destination table. Separate the columns with commas (,), for example, `"column":["id","name"]` . Open Search Writer can filter columns and change the order of columns. For example, an Open Search table has three columns: a, b, and c. If you want to write data only to columns c and b, you can set the column parameter in the format of `"column":["c","b"]` . During data synchronization, column a is automatically set to null.	Yes	N/A
batchSize	The number of data records to write at a time. Multiple data records are written to Open Search at a time. The advantage of Open Search is data query. The transactions per second (TPS) of Open Search is generally not high. Set this parameter based on the resources available for the account that is used to connect to Open Search. Generally, the size of a data record must be less than 1 MB, and the size of the data records to write at a time must be less than 2 MB.	Required only for writing data to a partitioned table	*300*
writeMode	The write mode. To ensure the idempotence of write operations, set the writeMode parameter to add/update when you configure Open Search Writer. • add: deletes the existing data record and inserts the new data record to Open Search, which is an atomic operation. • update: updates the existing data record based on the new data record, which is an atomic operation. **Note** Writing multiple data records to Open Search at a time is not an atomic operation. Part of the data may fail to be written. Exercise caution when you set the writeMode parameter. Open Search V3 does not support the update mode.	Yes	N/A

Parameter	Description	Required	Default value
ignoreWriteError	Specifies whether to ignore failed write operations. Example: `"ignoreWriteError":true` . If multiple data records are written to Open Search at a time, this parameter specifies whether to ignore failed write operations in the current batch. If you set the parameter to true, Open Search Writer continues to perform other write operations. If you set the parameter to false, the sync node ends and an error message is returned. We recommend that you use the default value.	No	*false*
version	The version of Open Search, for example, `"version":"v3"` . We recommend that you use Open Search V3 because the push operation faces many constraints in Open Search V2.	No	*v2*

Configure Open Search Writer by using the code editor

The following example shows how to configure a sync node to write data to Open Search. For more information, see Create a synchronization node by using the code editor.

```
{
    "type": "job",
    "version": "1.0",
    "configuration": {
        "reader": {},
        "writer": {
            "plugin": "opensearch",
            "parameter": {
                "accessId": "*********",
                "accessKey": "********",
                "host": "http://yyyy.aliyuncs.com",
                "indexName": "datax_xxx",
                "table": "datax_yyy",
                "column": [
                "appkey",
                "id",
                "title",
                "gmt_create",
                "pic_default"
                ],
                "batchSize": 500,
                "writeMode": add,
                "version":"v2",
                "ignoreWriteError": false
            }
        }
    }
}
```

5.3.21. Tablestore Writer

This topic describes the data types and parameters that are supported by Tablestore Writer and how to configure Tablestore Writer by using the codeless user interface (UI) and code editor.

Tablestore is a NoSQL database service that is built on the Apsara distributed operating system. The service allows you to store and access large volumes of structured data in real time. The data is stored in the tables that are created in Tablestore instances. Tablestore uses the data sharding and load balancing technologies to seamlessly expand the data scale.

Tablestore Writer connects to and writes data to the Tablestore server by using Tablestore SDK for Java. Tablestore Writer provides some features to optimize the write process, such as retry after write timeouts, retry after exceptions, and batch submission.

Tablestore Writer supports all Tablestore data types. The following table lists the data types that are supported by Tablestore Writer.

Category	Tablestore data type
Integer	INTEGER
Floating point	DOUBLE
String	STRING
Boolean	BOOLEAN
Binary	BINARY

Note To write data of the INTEGER type, set the data type to INT in the code editor. During data synchronization, Tablestore Writer converts the data from the INT data type to the INTEGER data type. If you set the data type to INTEGER, an error is reported in the log, and the synchronization node fails.

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
endPoint	The endpoint of the Tablestore server.	Yes	No default value
accessId	The AccessKey ID of the account that you use to connect to the Tablestore server.	Yes	No default value
accessKey	The AccessKey secret of the account that you use to connect to the Tablestore server.	Yes	No default value
instanceName	The name of the Tablestore instance. After you activate the Tablestore service, you must create an instance in the Tablestore console before you create and manage tables. Tablestore instances are the basic units that you can use to manage your Tablestore resources. Access control and resource measurement for applications are performed at the instance level.	Yes	No default value
table	The name of the table to which you want to write data. You can specify only one table. Data synchronization to multiple tables is not supported for Tablestore.	Yes	No default value

Parameter	Description	Required	Default value

primaryKey	The primary keys of the destination table in Tablestore. Specify the primary keys in a JSON array. Tablestore is a NoSQL database service. If you want to use Tablestore Writer to write data, you must specify the primary keys of the destination table.
	Data Integration supports data type conversion. Tablestore Writer can convert data from a data type other than STRING and INT to the STRING or INT data type. The following code provides a configuration example:
	`"primaryKey" : [`
	`    {"name":"pk1", "type":"string"},`
	`    {"name":"pk2", "type":"int"}`
	`                ],`
	Note
	• The primary keys in Tablestore must be of the STRING or INT type. Therefore, you must set the data type of a primary key to STRING or INT in the code editor.
	• You must set the primaryKey parameter to a value that is different from the value of the column parameter.
		Yes	No default value

column	The names of the columns to which you want to write data. Specify the names in a JSON array.
	Specify this parameter in the following format:
	`"column" : [`
	`{"name":"col2", "type":"INT"},`
	`{"name":"col3", "type":"STRING"}`
	`                ],`
	The name parameter specifies the name of the column to which you want to write. The type parameter specifies the data type of the column. Tablestore supports the following data types: STRING, INT, DOUBLE, BOOLEAN, and BINARY.
	Note
	You must set the primaryKey parameter to a value that is different from the value of the column parameter.
		Yes	No default value

writeMode	The write mode. Valid values: • *PutRow*: the PutRow API operation for Tablestore, which is used to insert data to a specific row. If the row does not exist, a new row is added. If the row exists, the row is overwritten. • *UpdateRow*: the UpdateRow API operation for Tablestore, which is used to update the data of a specific row. If the row does not exist, a new row is added. If the row exists, the values of the specified columns in the row are added, modified, or removed based on actual conditions.	Yes	No default value
requestTotalSizeLimitation	The maximum size of data that can be written to a single row in Tablestore. The parameter value must be of a numeric data type.	No	*1MB*

Parameter	Description	Required	Default value
attributeColumnSizeLimitation	The maximum size of data that can be written to a single attribute column in Tablestore. The parameter value is of a numeric data type.	No	*2MB*
primaryKeyColumnSizeLimitation	The maximum size of data that can be written to a single primary key column in Tablestore. The parameter value is of a numeric data type.	No	*1KB*
attributeColumnMaxCount	The maximum number of attribute columns to which data can be written in Tablestore. The parameter value is of a numeric data type.	No	*1,024*

Configure Tablestore Writer by using the codeless UI

This method is not supported.

Configure Tablestore Writer by using the code editor

In the following code, a synchronization node is configured to write data to Tablestore. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

Notice Delete the comments from the following code before you run the code.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"ots",// The writer type.
            "parameter":{
                "datasource":"",// The name of the data source.
                "column":[// The names of the columns to which you want to write data.
                    {
                        "name":"columnName1",// The name of a column.
                        "type":"INT"// The data type of a column.
                    },
                    {
                        "name":"columnName2",
                        "type":"STRING"
                    },
                    {
                        "name":"columnName3",
                        "type":"DOUBLE"
                    },
                    {
                        "name":"columnName4",
                        "type":"BOOL"
                    },
                    {
                        "name":"columnName5",
                        "type":"BINARY"
                    }
                ],
                "writeMode":"",// The write mode.
                "table":"",// The name of the table to which you want to write data.
                "primaryKey":[// The primary key of the destination table in Tablestore.
                "primaryKey":[// The primary keys of the destination table in Tablestore.
                    {
                        "name":"pk1",
                        "type":"STRING"
                    },
                    {
                        "name":"pk2",
                        "type":"INT"
                    }
                ]
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.22. Stream Writer

This topic describes the parameters that are supported by Stream Writer and how to configure Stream Writer by using the codeless user interface (UI) and code editor.

Stream Writer displays the data that is obtained from a reader on the screen or discards the data. Stream Writer is used to test the performance and basic features of Data Integration.

Parameters

print

- Description: specifies whether to display the data that is obtained from a reader on the screen.
- Required: No
- Default value: *true*

Configure Stream Writer by using the codeless UI

This method is not supported.

Configure Stream Writer by using the code editor

You can configure Stream Writer by using the code editor. For more information, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to display the data that is obtained from a reader on the screen:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",// The writer type.
            "parameter":{
                "print":false,// Specifies whether to display data on the screen.
                "fieldDelimiter":","// The column delimiter.
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.23. HybridDB for MySQL Writer

This topic describes the data types and parameters that are supported by HybridDB for MySQL Writer and how to configure HybridDB for MySQL Writer by using the codeless user interface (UI) and code editor.

HybridDB for MySQL Writer can write data to tables stored in HybridDB for MySQL databases. HybridDB for MySQL Writer connects to a remote HybridDB for MySQL database by using Java Database Connectivity (JDBC) and executes the `INSERT INTO` or `REPLACE INTO` statement to write data to the HybridDB for MySQL database. HybridDB for MySQL uses the InnoDB engine so that data is written to the database in batches.

> ⓘ **Note** Before you configure HybridDB for MySQL Writer, you must add a HybridDB for MySQL data source. For more information, see Add a HybridDB for MySQL data source.

HybridDB for MySQL Writer is designed for data developers to write data from data warehouses to HybridDB for MySQL databases. HybridDB for MySQL Writer can also be used as a data migration tool by users such as database administrators. HybridDB for MySQL Writer obtains data from a reader.

Note A synchronization node that uses HybridDB for MySQL Writer must have at least the permissions to execute the `INSERT INTO` statement. Whether other permissions are required depends on the SQL statements that you specify in the preSql and postSql parameters when you configure the node.

Data types

HybridDB for MySQL Writer supports most HybridDB for MySQL data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by HybridDB for MySQL Writer.

Category	HybridDB for MySQL data type
Integer	INT, TINYINT, SMALLINT, MEDIUMINT, BIGINT, and YEAR
Floating point	FLOAT, DOUBLE, and DECIMAL
String	VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT
Date and time	DATE, DATETIME, TIMESTAMP, and TIME
Boolean	BOOLEAN
Binary	TINYBLOB, MEDIUMBLOB, BLOB, LONGBLOB, and VARBINARY

Parameters

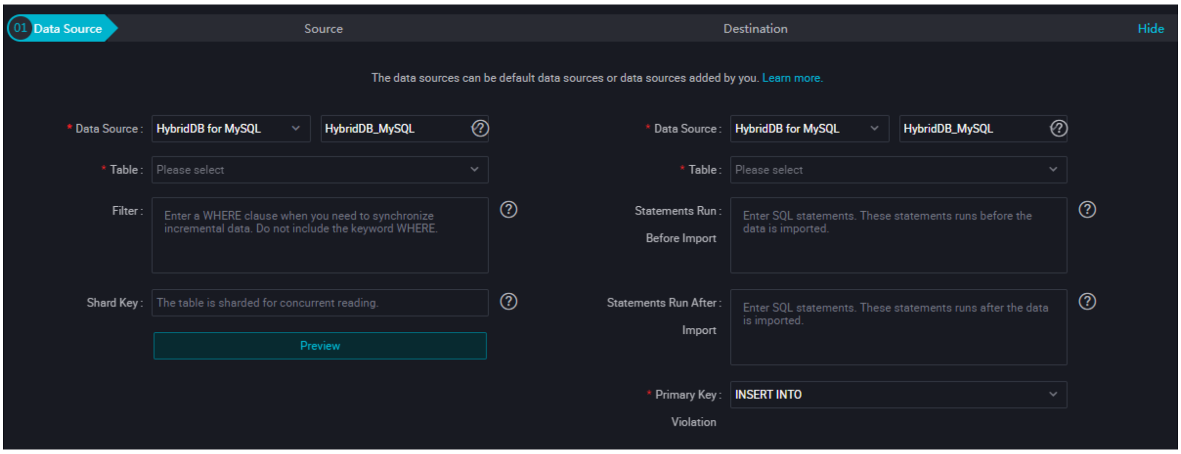
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table to which you want to write data.	Yes	No default value
writeMode	The write mode. Valid values: - *replace*: If no primary key conflict or unique index conflict occurs, data is processed in the way as that when you set this parameter to *insert*. If a conflict occurs, the specified fields in the rows in the destination table are updated. - *insert*: If no primary key conflict or unique index conflict occurs, data is directly written to the destination table. If a primary key conflict or unique index conflict occurs, data cannot be written to the conflicting rows, and the data that is not written to these rows is regarded as dirty data.	No	*insert*
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as `"column":["id","name","age"]`. If you want to write data to all the columns in the destination table, set the value to an asterisk (*), such as `"column":["*"]`.	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to delete outdated data. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value

Parameter	Description	Required	Default value
postSql	The SQL statement that you want to execute after the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to add a timestamp. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and HybridDB for MySQL and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

Configure HybridDB for MySQL Writer by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



The data sources can be default data sources or data sources added by you. Learn more.

Source:

* Data Source : HybridDB for MySQL HybridDB_MySQL

* Table : Please select

Filter : Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.

Shard Key : The table is sharded for concurrent reading.

Destination:

* Data Source : HybridDB for MySQL HybridDB_MySQL

* Table : Please select

Statements Run Before Import : Enter SQL statements. These statements runs before the data is imported.

Statements Run After Import : Enter SQL statements. These statements runs after the data is imported.

* Primary Key Violation : INSERT INTO

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	This parameter is equivalent to the table parameter that is described in the preceding section.
Statement Run Before Writing	The SQL statement that you want to execute before the synchronization node is run. This parameter is equivalent to the preSql parameter that is described in the preceding section.
Statement Run After Writing	The SQL statement that you want to execute after the synchronization node is run. This parameter is equivalent to the postSql parameter that is described in the preceding section.
Solution to Primary Key Violation	The write mode. This parameter is equivalent to the writeMode parameter that is described in the preceding section. You can select the desired write mode.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3.

Configure HybridDB for MySQL Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to write data to HybridDB for MySQL. For more information about the parameters, see the preceding parameter description.

```
{
    "type": "job",
    "steps": [
        {
            "parameter": {},
        {
            "parameter": {
                "postSql": [],// The SQL statement that you want to execute after the synchronization node is run.
                "datasource": "px_aliyun_hy***",// The name of the data source.
                "column": [// The names of the columns to which you want to write data.
                    "id",
                    "name",
                    "sex",
                    "salary",
                    "age",
                    "pt"
                ],
                "writeMode": "insert",// The write mode.
                "batchSize": 256,// The number of data records to write at a time.
                "encoding": "UTF-8",// The encoding format.
                "table": "person_copy",// The name of the table to which you want to write data.
                "preSql": []// The SQL statement that you want to execute before the synchronization node is run.
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",// The version number.
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {// The maximum number of dirty data records allowed.
            "record": ""
        },
        "speed": {
            "concurrent": 7,// The maximum number of parallel threads.
            "throttle": true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "mbps": 1,// The maximum transmission rate.
        }
    }
}
```

5.3.24. AnalyticDB for PostgreSQL Writer

This topic describes the data types and parameters that are supported by AnalyticDB for PostgreSQL Writer and how to configure AnalyticDB for PostgreSQL Writer by using the codeless user interface (UI) and code editor.

AnalyticDB for PostgreSQL Writer writes data to AnalyticDB for PostgreSQL databases. AnalyticDB for PostgreSQL Writer connects to a remote AnalyticDB for PostgreSQL database by using Java Database Connectivity (JDBC) and executes an SQL statement to write data to the AnalyticDB for PostgreSQL database.

Note Before you configure AnalyticDB for PostgreSQL Writer, you must configure an AnalyticDB for PostgreSQL data source. For more information, see Add an AnalyticDB for PostgreSQL data source.

Data types

AnalyticDB for PostgreSQL Writer supports most AnalyticDB for PostgreSQL data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by AnalyticDB for PostgreSQL Writer.

Data Integration data type	AnalyticDB for PostgreSQL data type
LONG	BIGINT, BIGSERIAL, INTEGER, SMALLINT, and SERIAL
DOUBLE	DOUBLE, PRECISION, MONEY, NUMERIC, and REAL
STRING	VARCHAR, CHAR, TEXT, BIT, and INET
DATE	DATE, TIME, and TIMESTAMP
BOOLEAN	BOOLEAN
BYTES	BYTEA

Note

- Data types that are not listed in the preceding table are not supported.
- The syntax such as `a_inet::varchar` is required when AnalyticDB for PostgreSQL Writer converts data to the MONEY, INET, or BIT data type.

Parameters

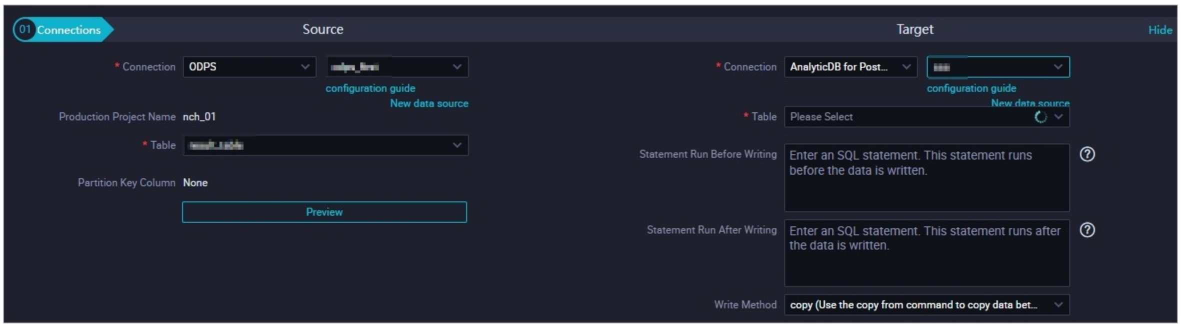
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table to which you want to write data.	Yes	No default value
writeMode	The write mode. Valid values: • insert: AnalyticDB for PostgreSQL Writer executes the `INSERT INTO...VALUES...` statement to write data to the AnalyticDB for PostgreSQL database. If a primary key conflict or unique index conflict occurs, data cannot be written to the conflicting rows, and the data that is not written to these rows is regarded as dirty data. We recommend that you select the insert mode. • copy: AnalyticDB for PostgreSQL provides the copy command to copy data between tables and the standard input or standard output file. Data Integration supports the `COPY FROM` statement, which allows you to copy data from a file to a table. We recommend that you use this mode if a performance issue occurs.	No	*insert*

Parameter	Description	Required	Default value
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as `"column":["id","name","age"]`. If you want to write data to all the columns in the destination table, set this parameter to an asterisk (*), such as "column":["*"].	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to delete outdated data. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to add a timestamp. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and AnalyticDB for PostgreSQL and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

Configure AnalyticDB for PostgreSQL Writer by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



01 Connections

Source

* Connection: ODPS [redacted]

Production Project Name: nch_01

* Table: [redacted]

Partition Key Column: None

Target

* Connection: AnalyticDB for Post... [redacted]

* Table: Please Select

Statement Run Before Writing: Enter an SQL statement. This statement runs before the data is written.

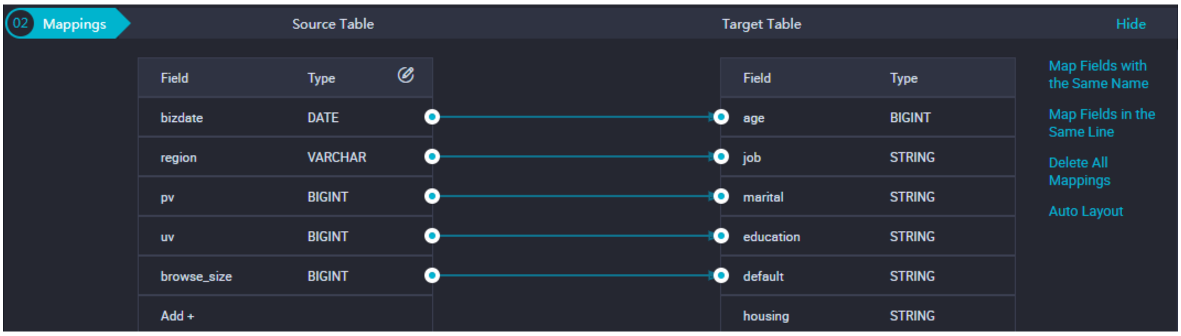
Statement Run After Writing: Enter an SQL statement. This statement runs after the data is written.

Write Method: copy (Use the copy from command to copy data bet...

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table to which you want to write data. This parameter is equivalent to the table parameter that is described in the preceding section.
Statement Run Before Writing	The SQL statement that you want to execute before the synchronization node is run. This parameter is equivalent to the preSql parameter that is described in the preceding section.
Statement Run After Writing	The SQL statement that you want to execute after the synchronization node is run. This parameter is equivalent to the postSql parameter that is described in the preceding section.

Parameter	Description
Write Method	The write mode. This parameter is equivalent to the writeMode parameter that is described in the preceding section. Valid values: **insert** and **copy**.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.



Field	Type	Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
Add +		housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3.

Configure AnalyticDB for PostgreSQL Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

ⓘ **Note** Delete the comments from the following code before you run the code.

```
{
    "type": "job",
    "steps": [
        {
            "parameter": {},
            "name": "Reader",
            "category": "reader"
        },
        {
            "parameter": {
                "postSql": [],// The SQL statement that you want to execute after the synchronization node is r
un.
                "datasource": "test_004",// The name of the data source.
                "column": [// The names of the columns to which you want to write data.
                    "id",
                    "name",
                    "sex",
                    "salary",
                    "age"
                ],
                "table": "public.person",// The name of the table to which you want to write data.
                "preSql": []// The SQL statement that you want to execute before the synchronization node is ru
n.
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",// The version number.
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {// The maximum number of dirty data records allowed.
            "record": ""
        },
        "speed": {
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":6, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    }
}
```

5.3.25. PolarDB Writer

This topic describes the data types and parameters that are supported by PolarDB Writer and how to configure PolarDB Writer by using the codeless user interface (UI) and code editor.

PolarDB Writer writes data to tables stored in PolarDB databases. PolarDB Writer connects to a remote PolarDB database by using Java Database Connectivity (JDBC) and executes an `INSERT INTO` or `REPLACE INTO` statement to write data to the PolarDB database. The PolarDB database must use the InnoDB engine because data is submitted to the PolarDB database in batches.

> Note Before you configure PolarDB Writer, you must configure a PolarDB data source. For more information, see Configure a PolarDB data source.

PolarDB Writer is designed for extract, transform, load (ETL) developers to import data in data warehouses to PolarDB databases. PolarDB Writer can also be used as a data migration tool by users such as database administrators. PolarDB Writer obtains data from a reader and writes the data to the destination database based on the value of the writeMode parameter.

> Note A synchronization node that uses PolarDB Writer must have at least the permissions to execute `INSERT INTO` and REPLACE INTO statements. Whether other permissions are required depends on the SQL statements that you specify in the preSql and postSql parameters when you configure the node.

Data types

Similar to PolarDB Reader, PolarDB Writer supports most PolarDB data types. Make sure that the data types of your database are supported.

The following table lists the data types that are supported by PolarDB Writer.

Category	PolarDB data type
Integer	INT, TINYINT, SMALLINT, MEDIUMINT, BIGINT, and YEAR
Floating point	FLOAT, DOUBLE, and DECIMAL
String	VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT, and LONGTEXT
Date and time	DATE, DATETIME, TIMESTAMP, and TIME
Boolean	BOOLEAN
Binary	TINYBLOB, MEDIUMBLOB, BLOB, LONGBLOB, and VARBINARY

Parameters

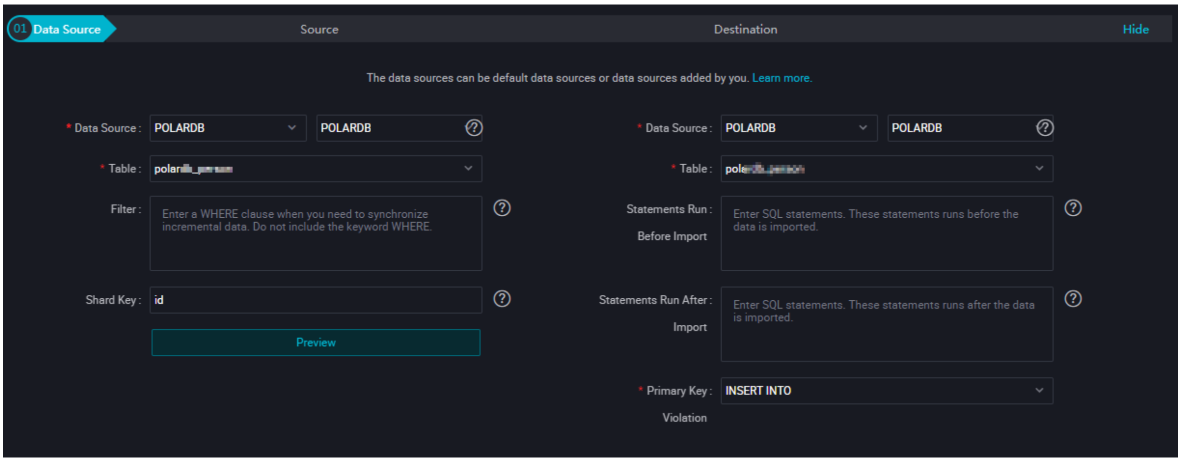
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table to which you want to write data.	Yes	No default value
writeMode	The write mode. Valid values: • replace: If no primary key conflict or unique index conflict occurs, data is processed in the same way as that when you set this parameter to insert. If a conflict occurs, rows in the destination table are deleted, and new rows are inserted. • insert: If no primary key conflict or unique index conflict occurs, data is directly written to the destination table. If a primary key conflict or unique index conflict occurs, data cannot be written to conflicting rows, and the data that is not written to these rows is regarded as dirty data. • update: If no primary key conflict or unique index conflict occurs, data is processed in the same way as that when you set this parameter to insert. If a conflict occurs, data in conflicting rows in the destination table is replaced by new data.	No	*insert*

Parameter	Description	Required	Default value
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as `"column": ["id", "name", "age"]` . If you want to write data to all the columns in the destination table, set this parameter to an asterisk (*), such as `"column": ["*"]` .	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to delete outdated data. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to add a timestamp. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor.	No	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and PolarDB and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

Configure PolarDB Writer by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



The data sources can be default data sources or data sources added by you. Learn more.

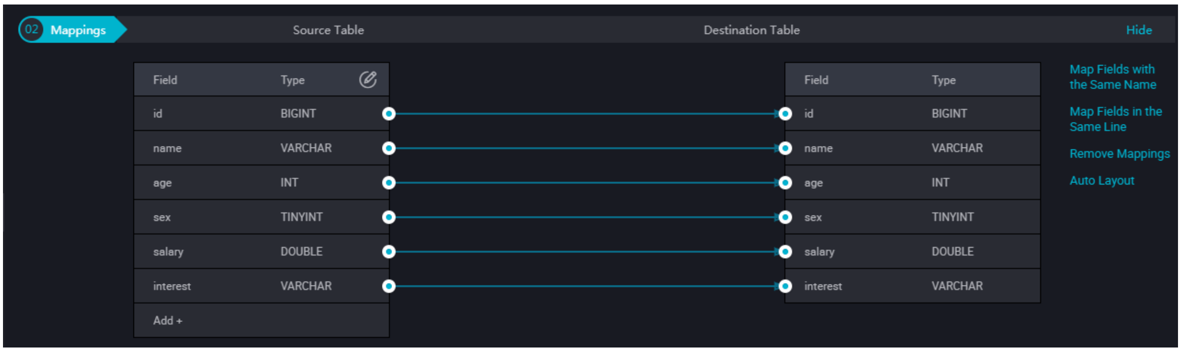
Source:
- * Data Source : POLARDB POLARDB
- * Table : [redacted]
- Filter : Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.
- Shard Key : id

Destination:
- * Data Source : POLARDB POLARDB
- * Table : [redacted]
- Statements Run Before Import : Enter SQL statements. These statements runs before the data is imported.
- Statements Run After Import : Enter SQL statements. These statements runs after the data is imported.
- * Primary Key Violation : INSERT INTO

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table to which you want to write data. This parameter is equivalent to the table parameter that is described in the preceding section.
Statement Run Before Writing	The SQL statement that you want to execute before the synchronization node is run. This parameter is equivalent to the preSql parameter that is described in the preceding section.

Parameter	Description
Statement Run After Writing	The SQL statement that you want to execute after the synchronization node is run. This parameter is equivalent to the postSql parameter that is described in the preceding section. Example: `update table set gmt_modify=now();` .
Solution to Primary Key Violation	The write mode. This parameter is equivalent to the writeMode parameter that is described in the preceding section.

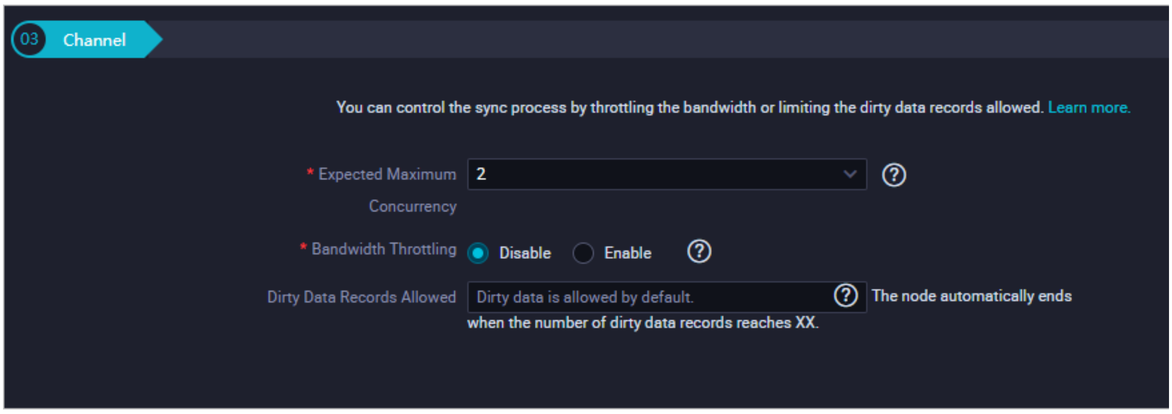
2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.



Field	Type	Field	Type
id	BIGINT	id	BIGINT
name	VARCHAR	name	VARCHAR
age	INT	age	INT
sex	TINYINT	sex	TINYINT
salary	DOUBLE	salary	DOUBLE
interest	VARCHAR	interest	VARCHAR

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specified rules.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: Dirty data is allowed by default. The node automatically ends when the number of dirty data records reaches XX.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.

Parameter	Description
Dirty Data Records Allowed	The maximum number of dirty data records allowed.

4.

Configure PolarDB Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to write data to PolarDB. For more information about the parameters, see the preceding parameter description.

```
{
    "type": "job",
    "steps": [
        {
            "parameter": {},
            "name": "Reader",
            "category": "reader"
        },
        {
            "parameter": {
                "postSql": [],// The SQL statement that you want to execute after the synchronization node is run.
                "datasource": "test_005",// The name of the data source.
                "column": [// The names of the columns to which you want to write data.
                    "id",
                    "name",
                    "age",
                    "sex",
                    "salary",
                    "interest"
                ],
                "writeMode": "insert",// The write mode.
                "batchSize": 256,// The number of data records to write at a time.
                "encoding": "UTF-8",// The encoding format.
                "table": "PolarDB_person_copy",// The name of the table to which you want to write data.
                "preSql": []// The SQL statement that you want to execute before the synchronization node is run.
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",// The version number.
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {// The maximum number of dirty data records allowed.
            "record": ""
        },
        "speed": {
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":6, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    }
}
```

5.3.26. TSDB Writer

TSDB Writer writes data points to Time Series Database (TSDB) that is developed by Alibaba Cloud. This topic describes the data types and parameters that are supported by TSDB Writer and how to configure TSDB Writer by using the codeless user interface (UI) and code editor.

Notice TSDB Writer supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.

Background information

TSDB is a high-performance, cost-effective, stable, and reliable online time series database service. TSDB features high read and write performance and provides a high compression ratio for data storage. TSDB also enables the interpolation and aggregation of time series data. TSDB can be used in various systems, such as IoT device monitoring systems, energy management systems (EMS) for enterprises, security monitoring systems for production, and electricity consumption monitoring systems.

You can write millions of data points to TSDB within seconds. TSDB provides the following features: high compression ratio, low-cost data storage, downsampling, interpolation, multi-dimensional aggregation, and visualized query results. These features help you resolve issues that are caused by a large number of data collection points on devices and frequent data collection. The issues include high storage costs and low write and query efficiency.

TSDB Writer connects to a TSDB instance by sending an HTTP request and writes data points by using the `/api/put` HTTP API endpoint.

Limits

- TSDB Writer supports TSDB V2.4.X and later.
- You can configure TSDB Writer only in the code editor. For more information about TSDB, see What is TSDB.

Data types

Category	Data Integration data type	TSDB data type
String	STRING	String to which a data point in TSDB is serialized. The data point can be a timestamp, metric, tag, or value.

Parameters

Parameter type	Parameter	Description	Required	Default value
Common parameter	sourceDbType	The type of the destination database.	No	*TSDB* Note Valid values: TSDB and RDB. The value TSDB indicates that the destination database is an OpenTSDB, Prometheus, or Timescale database. The value RDB indicates that the destination database is a relational database, such as a MySQL, Oracle, PostgreSQL, or Distributed Relational Database Service (DRDS) database.
	endpoint	The HTTP endpoint of the destination TSDB database. Specify the endpoint in the format of http://IP address:Port number.	Yes	No default value

Parameter type	Parameter	Description	Required	Default value
Parameters for TSDB	batchSize	The number of data records to write at a time. The value of this parameter is of the INT type and must be greater than 0.	No	*100*
	maxRetryTime	The maximum number of retries allowed after a failure. The value of this parameter is of the INT type and must be greater than 1.	No	*3*
	ignoreWriteError	Specifies whether to ignore write errors. The value of this parameter is of the BOOLEAN type. If you set this parameter to *true*, TSDB Writer continues to perform the write operation after a write error occurs. If the write operation fails after the specified number of retries, the synchronization node is terminated.	No	*false*
Parameters for RDB	endpoint	The HTTP endpoint of the destination relational database. Specify the endpoint in the format of http://IP address:Port number.	Yes	No default value
	column	The names of the columns to which you want to write data.	Yes	No default value **Note** You must specify the columns in the same order as the columns specified for a reader.
	columnType	The types of the columns in the relational database. The following types are supported: - timestamp: a timestamp column. - tag: a tag column. - metric_num: a metric column whose value is of a numeric data type. - metric_string: a metric column whose value is of a string data type.	Yes	No default value **Note** You must specify the columns in the same order as the columns specified for a reader.
	batchSize	The number of data records to write at a time. The value of this parameter is of the INT type and must be greater than 0.	No	*100*

Configure TSDB Writer by using the codeless UI

This method is not supported.

Configure TSDB Writer by using the code editor

In the following code, a synchronization node is configured to write data to a TSDB database. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "steps": [
        {
            "category": "reader",
            "name": "Reader",
            "parameter": {},
            "stepType": ""
        },
        {
            "category": "writer",
            "name": "Writer",
            "parameter": {
                "endpoint": "http://localhost:8242",
                "sourceDbType": "RDB",
                "batchSize": 256,
                "column": [
                    "name",
                    "type",
                    "create_time",
                    "price"
                ],
                "columnType": [
                    "tag",
                    "tag",
                    "timestamp",
                    "metric_num"
                ]
            },
            "stepType": "tsdb"
        }
    ],
    "type": "job",
    "version": "2.0"
}
```

Performance test report

- Characteristics of test data
  - Metric: a metric, which is m.
  - tag_k and tag_v: the key and value of a tag. The keys and values of the first four tags constitute a time series of 2,000,000 data points. The number of data points is calculated by using the following formula: `10 (zones) × 20 (clusters) × 100 (groups) × 100 (applications)`. The ip tag corresponds to the index of the 2,000,000 data points, starting from 1.

tag_k	tag_v
zone	z1 to z10
cluster	c1 to c20
group	g1 to g100
app	a1 to a100
ip	ip1 to ip2,000,000

  - value: a random value from 1 to 100.
  - interval: a collection interval of 10 seconds. The total duration of data collection is 3 hours, and a total number of 2,160,000,000 data points are collected. The number of data points is calculated by using the following formula: `3 × 60 × 60/10 × 2,000,000`.
- Performance test results

Number of channels	Data integration speed (record/s)	Data integration bandwidth (Mbit/s)
1	129,753	15.45
2	284,953	33.70
3	385,868	45.71

5.3.27. Configure AnalyticDB for MySQL 3.0 Writer

This topic describes the data types and parameters supported by AnalyticDB for MySQL 3.0 Writer and how to configure it by using the codeless user interface (UI) and code editor.

You must configure a connection before configuring AnalyticDB for MySQL 3.0 Writer.

Data types

The following table lists the data types supported by AnalyticDB for MySQL 3.0 Writer.

Category	AnalyticDB for MySQL 3.0 data type
Integer	INT, INTEGER, TINYINT, SMALLINT, and BIGINT
Floating point	FLOAT, DOUBLE, and DECIMAL
String	VARCHAR
Date and time	DATE, DATETIME, TIMESTAMP, and TIME
Boolean	BOOLEAN

Parameters

Parameter	Description	Required	Default value
datasource	The connection name. It must be identical to the name of the added connection. You can add connections in the code editor.	Yes	None
table	The name of the destination table.	Yes	None
writeMode	The write mode. Valid values: *insert into* and *replace into*. • *INSERT INTO*: If a primary key conflict or unique index conflict occurs, data cannot be written to the conflicting rows and is regarded as dirty data. • *REPLACE INTO*: If no primary key conflict or unique index conflict occurs, the action is the same as that of `INSERT INTO`. If a conflict occurs, original rows are deleted and new rows are inserted. That is, all fields of original rows are replaced.	No	*insert into*
column	The columns in the destination table to which data is written. Separate the columns with a comma (,). Example: `"column": ["id", "name", "age"]`. Set the value to an asterisk (*) if data is written to all the columns in the destination table. That is, set the column parameter as follows: `"column": ["*"]`. **Note** If the field name contains select, enclose the field name in grave accents (\` \`). Example: \`item_select_no\`.	Yes	None
preSql	The SQL statement to run before the sync node is run. For example, you can clear outdated data before data synchronization. Currently, you can run only one SQL statement on the codeless UI, and multiple SQL statements in the code editor. **Note** If you specify multiple SQL statements in the code editor, the system does not guarantee that they are run in the same transaction.	No	None
postSql	The SQL statement to run after the sync node is run. For example, you can add a timestamp after data synchronization. Currently, you can run only one SQL statement on the codeless UI, and multiple SQL statements in the code editor. **Note** If you specify multiple SQL statements in the code editor, the system does not guarantee that they are run in the same transaction.	No	None
batchSize	The number of data records to write at a time. Setting this parameter can greatly reduce the interactions between Data Integration and the AnalyticDB for MySQL 3.0 database over the network, and increase the throughput. However, an excessively large value may lead to the out of memory (OOM) error during the data synchronization process.	No	*1024*

Configure AnalyticDB for MySQL 3.0 Writer by using the codeless UI

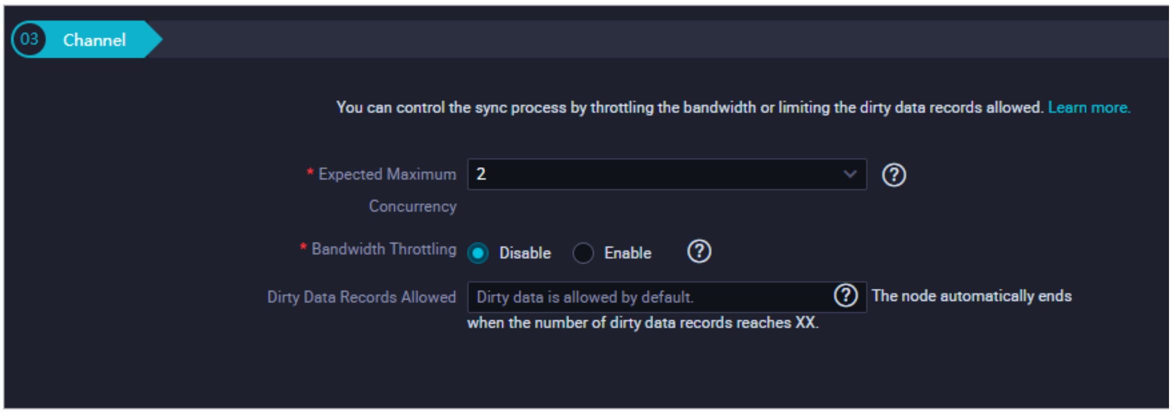
1. Configure the connections.

   Configure the source and destination connections for the sync node.

Parameter	Description
Connection	The datasource parameter in the preceding parameter description. Select a connection type, and enter the name of a connection that has been configured in DataWorks.
Table	The table parameter in the preceding parameter description.
Statement Run Before Writing	The preSql parameter in the preceding parameter description. Enter an SQL statement to run before the sync node is run.

Parameter	Description
Statement Run After Writing	The postSql parameter in the preceding parameter description. Enter an SQL statement to run after the sync node is run.
Solution to Primary Key Violation	The writeMode parameter in the preceding parameter description. Select the expected write mode.
Data Records per Write	The number of data records to write at a time. The batchSize parameter in the preceding parameter description. This parameter takes effect only when wirteMode is set to insert into.

2. Configure field mapping, that is, the column parameter in the preceding parameter description. Fields in the source table on the left have a one-to-one mapping with fields in the destination table on the right.
3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: Dirty data is allowed by default. The node automatically ends when the number of dirty data records reaches XX.

Parameter	Description
Expected Maximum Concurrency	The maximum number of concurrent threads to read and write data to data storage within the sync node. You can configure the concurrency for a node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Resource Group	The servers on which nodes are run. If an excessively large number of nodes are run in the default resource group, some nodes may be delayed due to insufficient resources. In this case, we recommend that you purchase an exclusive resource group for data integration or add a custom resource group. For more information, see DataWorks exclusive resources and Create and use a custom resource group for Data Integration.

Configure AnalyticDB for MySQL 3.0 Writer by using the code editor

In the following code, a node is configured to write data to an AnalyticDB for MySQL 3.0 database. For more information about the parameters, see the preceding parameter description.

```
{
    "type": "job",
    "steps": [
        {
            "stepType": "stream",
            "parameter": {},
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "analyticdb_for_mysql", // The writer type.
             "parameter": {
                "postSql": [], // The SQL statement to run after the sync node is run.
                 "tableType": null, // The reserved field. Default value: null.
                 "datasource": "hangzhou_ads", // The connection name.
                 "column": [ // The columns in the destination table to which data is written.
                     "id",
                    "value"
                ],
                "guid": null,
                "writeMode": "insert", // The write mode. For more information, see the description of the writeMode parameter.
                 "batchSize": 2048, // The number of data record to write at a time. For more information, see description of the batchSize parameter.
                 "encoding": "UTF-8", // The encoding format.
                 "table": "t5", // The name of the destination table.
                 "preSql": [] // The SQL statement to run before the sync node is run.
             },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0", // The version number.
     "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record":"0" // The maximum number of dirty data records allowed.
         },
        "speed": {
            "concurrent": 2, // The maximum number of concurrent threads.
             "throttle": false // Specifies whether to enable bandwidth throttling. A value of false indicates that the bandwidth is not throttled. A value of true indicates that the bandwidth is throttled. The maximum transmission rate takes effect only if you set this parameter to true.
         }
    }
}
```

5.3.28. GDB Writer

This topic describes the parameters that are supported by Graph Database (GDB) Writer and how to configure GDB Writer by using the codeless user interface (UI) and code editor.

GDB is a real-time and reliable online database service that uses the Property Graph model to process highly connected data queries and store the processed data. GDB uses Apache TinkerPop Gremlin as the query language, which allows you to build queries that can navigate highly connected datasets with improved efficiency.

Notice

- GDB Writer supports only exclusive resource groups for Data Integration, but not the default resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Add a custom resource group for Data Integration.
- You must separately configure data integration tasks for a vertex and an edge because the settings of the two tasks are different.

Limits

- You must run a synchronization node to synchronize vertex data before you run a synchronization node to synchronize edge data.
- Limits on vertices:
  - A vertex must have a name, which is specified by the label parameter.
  - A vertex must have a unique primary key of the STRING type. If the primary key is not a string, GDB Writer forcibly converts the primary key into a string.
  - Exercise caution when you set the idTransRule parameter. If you set this parameter to *none*, make sure that the primary key of each vertex is unique among all vertices.
- Limits on edges:
  - An edge must have a name, which is specified by the label parameter.
  - A primary key is optional for an edge.
    - If you specify a primary key for an edge, make sure that the primary key is unique among all edges.
    - If you do not specify a primary key for an edge, GDB Writer automatically generates a universally unique identifier (UUID) of the STRING type for the edge. If the UUID is not a string, GDB Writer forcibly converts the UUID into a string.
  - Exercise caution when you set the idTransRule parameter. If you set this parameter to *none*, make sure that the primary key of each edge is unique among all edges.
  - The srcIdTransRule and dstIdTransRule parameters are required for an edge. The values of the two parameters must be the same as the value of the idTransRule parameter of the related vertex.
- Unless otherwise specified, field names and enumerated values in this topic are case-sensitive.
- GDB Writer supports only the UTF-8 encoding format. Source data must be encoded in UTF-8.
- Due to network constraints, synchronization nodes that are used to synchronize data to GDB databases must be run by using exclusive resource groups for Data Integration. You must purchase an exclusive resource group for Data Integration and associate the group with the virtual private cloud (VPC) in which the GDB instance resides in advance. For more information, see Exclusive resource group mode. Scheduling nodes can be run by using the default resource group.

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
label	The label, which is the name of the vertex or edge. GDB Writer can obtain labels from columns in the source table. For example, if you set this parameter to ${0}, GDB Writer uses the value of the first column as the label. The column index starts from 0.	Yes	No default value
labelType	The type of the label. Valid values: • VERTEX: a vertex. • EDGE: an edge.	Yes	No default value

Parameter	Description	Required	Default value
srcLabel	- The name of the start vertex in an edge when the labelType parameter is set to EDGE. In this case, this parameter can be left empty if srcIdTransRule is set to *none*. If srcIdTransRule is set to another value, this parameter is required. - Leave this parameter empty if the labelType parameter is set to VERTEX.	No	No default value
dstLabel	- The name of the end vertex in an edge when the labelType parameter is set to EDGE. In this case, this parameter can be left empty if dstIdTransRule is set to *none*. If dstIdTransRule is set to another value, this parameter is required. - Leave this parameter empty if the labelType parameter is set to VERTEX.	No	No default value
writeMode	The mode in which GDB Writer processes data records with duplicate primary keys. Valid values: - *INSERT*: returns an error message. The number of error data records is increased by 1. - *MERGE*: overrides the existing data record with the new one.	Yes	*INSERT*
idTransRule	The rule for converting the primary key. Valid values: - *labelPrefix*: converts the primary key into the `{label}-{column in source}` format. - *none*: does not convert the primary key.	Yes	*none*
srcIdTransRule	The rule for converting the primary key of the start vertex when the labelType parameter is set to EDGE. Valid values: - *labelPrefix*: converts the primary key into the `{label}-{column in source}` format. - *none*: does not convert the primary key. In this case, the srcLabel parameter can be left empty.	Required when the labelType parameter is set to EDGE	*none*
dstIdTransRule	The rule for converting the primary key of the end vertex when the labelType parameter is set to EDGE. Valid values: - *labelPrefix*: converts the primary key into the `{label}-{column in source}` format. - *none*: does not convert the primary key. In this case, the dstLabel parameter can be left empty.	Required when the labelType parameter is set to EDGE	*none*

Parameter	Description	Required	Default value

column	The vertices or edges that you want to synchronize.
	• name: the name of the vertex or edge property.
	• value: the value of the vertex or edge property. You can customize a value only in the code editor.
	◦ *${N}*: uses the value of the Nth column in the source as the value of the vertex or edge property. N indicates the column index, which starts from 0.
	◦ *${0}*: uses the value of the first column in the source as the value of the vertex or edge property.
	◦ *test-${0}*: appends a fixed string such as test- to the beginning or end of *${0}*.
	◦ *${0}-${1}*: combines the values of multiple columns in the source as the value of a vertex or edge property. You can also add fixed strings at any positions, such as `test-${0}-test1-${1}-test2` .
	• *type*: the data type of the vertex or edge property.
	The primary key must be of the STRING type. If the value obtained from the source is not a string, GDB Writer forcibly converts the value into a string. Make sure that the value can be converted into a string.
	Other properties can be of the INT, LONG, FLOAT, DOUBLE, BOOLEAN, or STRING type.
	• *columnType*: the category of the vertex or edge property that you want to synchronize.
	◦ For both vertices and edges
	primaryKey: the primary key.
	◦ For vertices
	▪ *vertexProperty*: a common property of a vertex.
	▪ *vertexJsonProperty*: a JSON property of the vertex. For more information about the value structure, see the sample of properties.
	◦ For edges
	▪ *srcPrimaryKey*: the primary key of the start vertex.
	▪ *dstPrimaryKey*: the primary key of the end vertex.
	▪ *edgeProperty*: a common property of an edge.
	▪ *edgeJsonProperty*: a JSON property of an edge. For more information about the value structure, see the sample of properties.
	Sample of properties
	`{"properties":[`
	`    {"k":"name","t":"string","v":"tom"},`
	`    {"k":"age","t":"int","v":"20"},`
	`    null`
	`]}`
		Yes	*No default value*

Configure GDB Writer by using the codeless UI

This method is not supported.

Configure GDB Writer by using the code editor

In the following code, a synchronization node is configured to write data to a GDB database by using the code editor. For more information, see Create a synchronization node by using the code editor.

- Configure a synchronization node to write data about vertices to a GDB database

```
{
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    },
    "setting":{
        "errorLimit":{
            "record":"100" // The maximum number of dirty data records allowed.
        },
        "jvmOption":"",
        "speed":{
             "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates t
hat bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The
mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":3, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "steps":[
        {
            "category":"reader",
            "name":"Reader",
            "parameter":{
                "column":[
                    "*"
                ],
                "datasource":"_ODPS",
                "emptyAsNull":true,
                "guid":"",
                "isCompress":false,
                "partition":[],
                "table":""
            },
            "stepType":"odps"
        },
        {
            "category":"writer",
            "name":"Writer",
            "parameter": {
                "datasource": "testGDB", // The name of the data source.
                "label": "person", // The label, which is the name of the vertex.
                "srcLabel": "", // You do not need to specify this parameter for a vertex.
                "dstLabel": "", // You do not need to specify this parameter for a vertex.
                "labelType": "VERTEX", // The type of the label. VERTEX indicates a vertex.
                "writeMode": "INSERT", // The mode in which GDB Writer processes data records with duplicate
primary keys.
                "idTransRule": "labelPrefix", // The rule for converting the primary key of a vertex.
                "srcIdTransRule": "none", // You do not need to specify this parameter for a vertex.
                "dstIdTransRule": "none", // You do not need to specify this parameter for a vertex.
                "column": [
                    {
                        "name": "id", // The name of the column.
                        "value": "${0}", // The value of the first column in the source is used as the value
of the vertex property. If multiple columns are specified, the columns can be concatenated. In this example,
0 is the column index.
                        "type": "string", // The data type of the column.
                        "columnType": "primaryKey" // The category of the vertex property. The value primaryK
ey indicates the primary key.
                    }, // The primary key of the vertex. The value must be an ID of the STRING type, and the
record must exist.
                    {
                        "name": "person_age",
                        "value": "${1}", // The value of the second column in the source is used as the value
of the vertex property. If multiple columns are specified, the columns can be concatenated.
                        "type": "int",
                        "columnType": "vertexProperty" // The category of the vertex property. This parameter
indicates a common vertex property.
                    }, // A common property of the vertex. The value can be of the INT, LONG, FLOAT, DOUBLE,
BOOLEAN, or STRING type.
                    {
                        "name": "person_credit",
                        "value": "${2}", // The value of the third column in the source is used as the value
of the vertex property. If multiple columns are specified, the columns can be concatenated.
                        "type": "string",
                        "columnType": "vertexProperty"
                    }, // A common property of the vertex.
                ]
            }
            "stepType":"gdb"
        }
    ],
    "type":"job",
    "version":"2.0"
}
```

- Configure a synchronization node to write data about edges to a GDB database

```
{
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    },
    "setting":{
        "errorLimit":{
            "record":"100"// The maximum number of dirty data records allowed.
        },
        "jvmOption":"",
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates th
at bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The m
bps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":3, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "steps":[
        {
            "category":"reader",
            "name":"Reader",
            "parameter":{
                "column":[
                    "*"
                ],
                "datasource":"_ODPS",
                "emptyAsNull":true,
                "guid":"",
                "isCompress":false,
                "partition":[],
                "table":""
            },
            "stepType":"odps"
        },
        {
            "category":"writer",
            "name":"Writer",
            "parameter": {
                "datasource": "testGDB", // The name of the data source.
                "label": "use", // The label, which is the name of the edge.
                "labelType": "EDGE", // The type of the label. EDGE indicates an edge.
                "srcLabel": "person", // The name of the start vertex.
                "dstLabel": "software", // The name of the end vertex.
                "writeMode": "INSERT", // The mode in which GDB Writer processes data records with duplicate
primary keys.
                "idTransRule": "labelPrefix", // The rule for converting the primary key of an edge.
                "srcIdTransRule": "labelPrefix", // The rule for converting the primary key of the start vert
ex.
                "dstIdTransRule": "labelPrefix", // The rule for converting the primary key of the end vertex
.
                "column": [
                    {
                        "name": "id", // The name of the column.
                        "value": "${0}", // The value of the first column in the source is used as the value
of the edge property. If multiple columns are specified, they can be concatenated.
                        "type": "string", // The data type of the column.
                        "columnType": "primaryKey" // The category of the edge property. The value primaryKey
indicates the primary key.
                    }, // The primary key of the edge. The value must be an ID of the STRING type, and the re
cord must exist.
                    {
                        "name": "id",
                        "value": "${1}", // The value of the second column in the source is used as the value
of the edge property. If multiple columns are specified, they can be concatenated. The mapping rule must be t
he same as that configured when you import the vertex.
                        "type": "string",
                        "columnType": "srcPrimaryKey" // The category of the edge property. This parameter in
dicates the primary key of the start vertex.
                    }, // The primary key of the start vertex. The value must be an ID of the STRING type, an
d the record must exist.
                    {
                        "name": "id",
                        "value": "${2}", // The value of the third column in the source is used as the value
of the edge property. If multiple columns are specified, they can be concatenated. The mapping rule must be t
he same as that configured when you import the vertex.
                        "type": "string",
                        "columnType": "dstPrimaryKey" // The category of the edge property. This parameter in
dicates the primary key of the end vertex.
                    }, // The primary key of the end vertex. The value must be an ID of the STRING type, and
the record must exist.
                    {
                        "name": "person_use_software_time",
                        "value": "${3}", // The value of the fourth column in the source is used as the value
of the edge property. If multiple columns are specified, they can be concatenated.
                        "type": "long",
                        "columnType": "edgeProperty" // The category of the edge property. edgeProperty indic
ates a common edge property.
                    }, // A common property of the edge. The value can be of the INT, LONG, FLOAT, DOUBLE, BO
OLEAN, or STRING type.
                    {
                        "name": "person_regist_software_name",
                        "value": "${4}", // The value of the fifth column in the source is used as the value
of the edge property. If multiple columns are specified, they can be concatenated.
                        "type": "string",
                        "columnType": "edgeProperty"
                    }, // A common property of the edge.
                    {
                        "name": "id",
                        "value": "${5}", // The value of the sixth column in the source is used as the value
of the edge property. If multiple columns are specified, they can be concatenated.
                        "type": "long",
                        "columnType": "edgeProperty"
                    }, // A common property of the edge. The value is an ID. Different from the primary key,
this property is optional.
                ]
            }
            "stepType":"gdb"
        }
    ],
    "type":"job",
    "version":"2.0"
}
```

5.3.29. MaxCompute Writer

This topic describes the parameters that are supported by MaxCompute Writer and how to configure MaxCompute Writer by using the codeless user interface (UI) and code editor.

Prerequisites

Before you configure MaxCompute Writer, you must configure a MaxCompute data source. For more information, see Add a MaxCompute data source.

Background information

MaxCompute Writer is designed for developers to insert data to or update data in MaxCompute. MaxCompute Writer can write gigabytes or terabytes of data to MaxCompute. For more information about MaxCompute, see What is MaxCompute?.

MaxCompute Writer writes data to MaxCompute by using Tunnel commands based on the information you specified, such as the source project, table, partition, and field. For more information about common Tunnel commands, see Tunnel commands.

For a table with a strict schema, such as a table in a MySQL database or MaxCompute project, Data Integration reads data from the table and stores the data in the memory. Then, Data Integration converts the data to the format that is supported by the destination and writes the data to the destination.

If the data conversion fails or the data fails to be written to the destination, the data is regarded as dirty data. You can specify the maximum number of dirty data records allowed.

> **Note** If the data in the source contains a null value, MaxCompute Writer cannot convert the data to the VARCHAR type.

Parameters

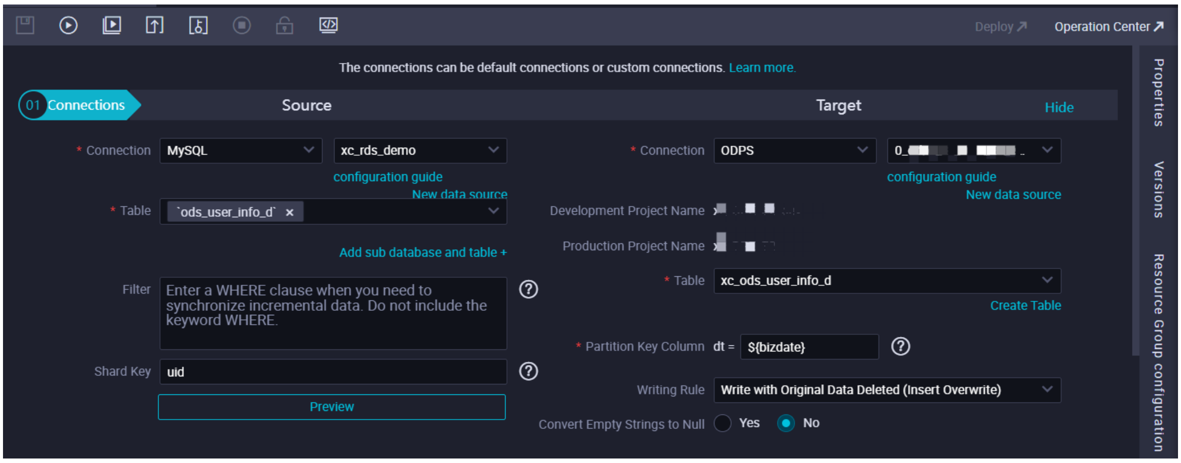
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table to which you want to write data. The name is not case-sensitive. You can specify only one table.	Yes	No default value

Parameter	Description	Required	Default value
partition	The partition to which data is written. The last-level partition must be specified. For example, if you want to write data to a table with three-level partitions, set the partition parameter to a value that contains the third-level partition information, such as `pt=20150101, type=1, biz=2` . • To write data to a non-partitioned table, do not configure this parameter. The data is directly written to the destination table. • MaxCompute Writer does not support data write operations based on the partition route. To write data to a partitioned table, make sure that the data is written to the lowest-level partition.	Required only for partitioned tables	No default value
column	The names of the columns to which you want to write data. If you want to write data to all the columns in the destination table, set this parameter to an asterisk (*), such as `"column": ["*"]` . If you want to write data only to specific columns in the destination table, set this parameter to the names of the columns. Separate the names with commas (,), such as `"column":["id","name"]` . • MaxCompute Writer can filter columns and change the order of columns. For example, a MaxCompute table has three columns: a, b, and c. If you want to write data only to column c and column b, you can enter `"column": ["c","b"]` . During data synchronization, column a is automatically set to null. • The column parameter must explicitly specify all the columns to which you want to write data. This parameter cannot be left empty.	Yes	No default value
truncate	To ensure the idempotence of write operations, set the `truncate` parameter to true. If a failed synchronization node is rerun due to a write failure, MaxCompute Writer deletes the data that has been written to the destination table and writes the source data again. This ensures that the same data is written for each rerun. MaxCompute Writer uses MaxCompute SQL to delete data. MaxCompute SQL cannot ensure data atomicity. Therefore, the TRUNCATE operation is not an atomic operation. Conflicts may occur when multiple nodes delete data from the same table or partition in parallel. To prevent this issue, we recommend that you do not execute multiple DDL statements to write data to the same partition at the same time. You can create different partitions for nodes that need to run in parallel.	Yes	No default value

Configure MaxCompute Writer by using the codeless UI

1. Configure data sources.

Configure **Source** and **Target** for the synchronization node.



The connections can be default connections or custom connections. Learn more.

Source:

- * Connection: MySQL, xc_rds_demo
- * Table: `ods_user_info_d` ×
- Filter: Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.
- Shard Key: uid

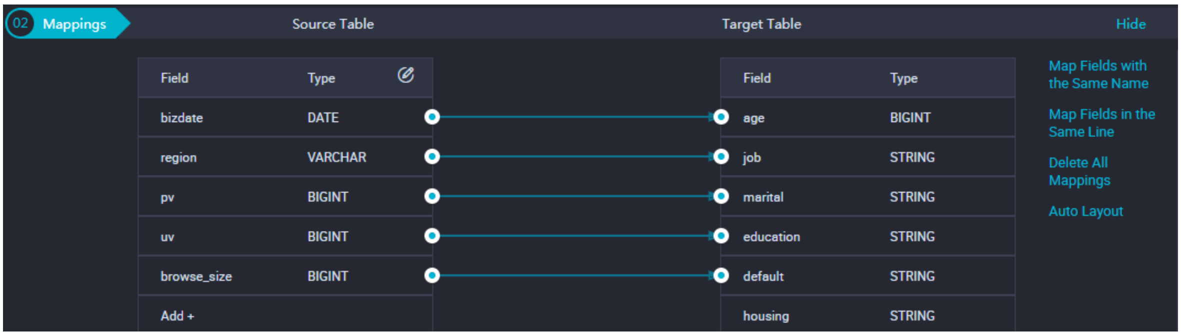
Target:

- * Connection: ODPS, [redacted]
- Development Project Name: [redacted]
- Production Project Name: [redacted]
- * Table: xc_ods_user_info_d
- * Partition Key Column: dt = ${bizdate}
- Writing Rule: Write with Original Data Deleted (Insert Overwrite)
- Convert Empty Strings to Null: ○ Yes ● No

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table to which you want to write data. This parameter is equivalent to the table parameter that is described in the preceding section. If the table is a partitioned table, you must specify the partition to which you want to write data. You can use scheduling parameters when you specify the partition. For more information about scheduling parameters, see Overview of scheduling parameters.
Writing Rule	The write rule. Valid values: ◦ **Write with Original Data Deleted (Insert Overwrite)**: All data in the table or partition is deleted before MaxCompute Writer writes data. This rule is equivalent to the `INSERT OVERWRITE` statement. ◦ **Write with Original Data Retained (Insert Into)**: No data is deleted before MaxCompute Writer writes data. New data is appended upon each run. This rule is equivalent to the `INSERT INTO` statement. Note ◦ MaxCompute Reader reads data by using Tunnel commands. Synchronization nodes cannot filter data. Each synchronization node reads all the data from a table or partition. ◦ MaxCompute Writer writes data by using Tunnel commands instead of the INSERT INTO statement. You can view complete data in the destination table only after the synchronization node is successfully run. Pay attention to the node dependencies.
Convert Empty Strings to Null	Specifies whether to convert empty strings to null.

Parameter	Description
Allow Query After Synchronization is Complete	Data write is a process that lasts for a period of time. This parameter specifies whether to allow users to query synchronized data only after the data synchronization node finishes running. Note The setting of this parameter does not take effect for nodes for which the **Distributed Execution** feature is enabled. ◦ If you select **Yes**, you can query the data that is synchronized to MaxCompute by the data synchronization node only after the node finishes running. Note A data synchronization node can synchronize a maximum of 1 TB of data to a destination. If severe data skew occurs, the maximum of amount of data that is written to the destination may be less than this threshold. ◦ If you select **No**, you can query the data that is synchronized to MaxCompute by the data synchronization node when the node is running. Note Before source data is written to MaxCompute, a partition may be automatically created in the destination MaxCompute table or existing data in the destination MaxCompute table or destination partition may be deleted. Therefore, when the data synchronization node is running, you may view the partition that is automatically created or the table or partition from which existing data is deleted regardless of whether you set Allow Query After Synchronization is Complete to Yes or No. We recommend that you do not determine whether data synchronization is complete based on whether the automatically created partition exists in the destination or no more data records are written to the destination. If you want to determine whether data synchronization is complete, we recommend that you create an ODPS SQL node and configure the ODPS SQL node as the descendant node of your data synchronization node. The system creates a MaxCompute table or a partition for the ODPS SQL node. You can determine whether data synchronization is complete based on whether the MaxCompute table or partition exists in the ODPS SQL node.

2. Configure field mappings. This operation is equivalent to setting the column parameter when you use the code editor. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.



Field	Type	Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
Add +		housing	STRING

Parameter	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3.

Configure MaxCompute Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to write data to MaxCompute. For more information about the parameters, see the preceding parameter description.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"odps",// The writer type.
            "parameter":{
                "partition":"",// The name of the partitions to which you want to write data.
                "truncate":true,// The write rule.
                "compress":false,// Specifies whether to enable compression.
                "datasource":"odps_first",// The name of the data source.
            "column": [// The names of the columns to which you want to write data.
                "id",
                "name",
                "age",
                "sex",
                "salary",
                "interest"
                ],
                "emptyAsNull":false,// Specifies whether to convert empty strings to null.
                "table":""// The name of the table to which you want to write data.
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

If you want to specify the Tunnel endpoint, you can configure the data source in the code editor. To configure the data source, replace `"datasource":"",` in the preceding code with detailed parameters of the data source. Example:

```
"accessId":"<yourAccessKeyId>",
 "accessKey":"<yourAccessKeySecret>",
 "endpoint":"http://service.eu-central-1.maxcompute.aliyun-inc.com/api",
 "odpsServer":"http://service.eu-central-1.maxcompute.aliyun-inc.com/api",
"tunnelServer":"http://dt.eu-central-1.maxcompute.aliyun.com",
"project":"**********",
```

Additional information

- Column filter

  MaxCompute Writer allows you to perform operations that MaxCompute does not support, such as filtering columns, reordering columns, and setting empty fields to null. If you want to write data to all the columns in the destination table, set the column parameter to `["*"]`.

  For example, a MaxCompute table has three columns: a, b, and c. If you want to write data only to column c and column b, you can set the column parameter to `["c","b"]`. The first column and the second column in the source table are written to column c and column b in the MaxCompute table. During data synchronization, column a is automatically set to null.

- Handling column configuration errors

  To prevent data loss caused by redundant columns and ensure high data reliability, MaxCompute Writer returns an error message if the number of columns that are to be written is more than that in the destination table. For example, if a MaxCompute table contains columns a, b, and c, MaxCompute Writer returns an error message if more than three columns are to be written to the table.

- Partition configuration

  MaxCompute Writer can write data to the lowest-level partition but cannot write data to a specified partition based on a field. To write data to a partitioned table, specify the lowest-level partition. For example, if you want to write data to a table with three-level partitions, set the partition parameter to a value that contains the third-level partition information, such as `pt=20150101, type=1, biz=2`. The data cannot be written if you set the partition parameter to `pt=20150101, type=1` or `pt=20150101`.

- Node rerunning

  To ensure the idempotence of write operations, set the `truncate` parameter to true. If a failed synchronization node is rerun due to a write failure, MaxCompute Writer deletes the data that has been written to the destination table and writes the source data again. This ensures that the same data is written for each rerun. If a synchronization node is interrupted due to other exceptions, the idempotence of data cannot be ensured, the data cannot be rolled back, and the node cannot be automatically rerun. You can ensure the idempotence of write operations and the data integrity by setting the truncate parameter to true.

  > ? **Note** If the truncate parameter is set to true, all data in the specified partition or table is deleted before a rerun. Exercise caution when you configure this parameter.

5.3.30. Hive Writer

Hive Writer writes data to Hadoop Distributed File System (HDFS) and loads the data to Hive. This topic describes how Hive Writer works, the parameters that are supported by Hive Writer, and how to configure Hive Writer by using the codeless user interface (UI) and code editor.

Background information

Hive is a Hadoop-based data warehouse tool that is used to process large amounts of structured logs. Hive maps structured data files to a table and allows you to execute SQL statements to query data in the table.

Notice Hive Writer supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration, Use the default resource group, and Create a custom resource group for Data Integration. For more information about the Hive versions that are supported by Hive Writer, see the Hive versions supported by Hive Writer section of this topic.

Hive converts Hibernate Query Language (HQL) or SQL statements into MapReduce programs.

- Hive stores processed data in HDFS.
- Hive uses MapReduce programs to analyze data at the underlying layer.
- Hive runs MapReduce programs on Yarn.

How it works

Hive Writer connects to a Hive metastore and obtains the storage path, format, and column delimiter of the HDFS file to which you want to write data. Then, Hive Writer writes data to the HDFS file and loads the data in the HDFS file to the destination Hive table by using Java Database Connectivity (JDBC).

The underlying logic of Hive Writer is the same as that of HDFS Writer. You can set parameters for HDFS Writer in the parameters of Hive Writer. Data Integration transparently transmits the configured parameters to HDFS Writer.

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source.	Yes	N/A
column	The names of the columns to which you want to write data, such as `"column": ["id", "name"]` . - You can write data to some of the columns. - The column parameter must be set to all the columns to which you want to write data. This parameter cannot be left empty. - The order of the columns cannot be changed.	Yes	N/A
table	The name of the Hive table to which you want to write data. **Note** The name is case-sensitive.	Yes	N/A
partition	The partition information in the Hive table. - If you want to write data to a partitioned Hive table, this parameter is required. After you specify this parameter, Hive Writer writes data to the partition that is specified by this parameter. - If you want to write data to a non-partitioned table, this parameter is not required.	No	N/A

Parameter	Description	Required	Default value
writeMode	The mode in which Hive Writer loads data to the Hive table. After data is written to an HDFS file, Hive Writer executes the `LOAD DATA INPATH (overwrite) INTO TABLE` statement to load data to the Hive table. This parameter specifies the mode in which Hive Writer loads data from the HDFS file to the Hive table. Valid values: - *truncate*: Hive Writer deletes existing data before it loads data to the Hive table. - *append*: Hive Writer retains the existing data and appends data to the Hive table. - Other: Hive Writer writes data to the HDFS file but does not load the data to the Hive table. **Note** Set this parameter with caution. Pay attention to the destination directory and the value of this parameter to prevent data from being unexpectedly deleted. This parameter and the hiveConfig parameter must be used in pairs.	Yes	N/A

hiveConfig	The extended parameters for Hive. Valid values: hiveCommand, jdbcUrl, username, and password.
	- hiveCommand: the full path of the Hive client. After you run the `hive -e` command, the `LOAD DATA INPATH` statement is executed to load data based on the mode that is specified by the writeMode parameter.
	The client that is specified by the hiveCommand parameter provides access information about Hive.
	- jdbcUrl, username, and password: the information that is required to connect to Hive by using JDBC. After Hive Writer connects to Hive by using JDBC, Hive Writer executes the `LOAD DATA INPATH` statement to load data based on the mode that is specified by the writeMode parameter.
	`"hiveConfig": {`
	`    "hiveCommand": "",`
	`    "jdbcUrl": "",`
	`    "username": "",`
	`    "password": ""`
	`        }`
	- Hive Writer uses an HDFS client to write data to HDFS files. You can use this parameter to specify advanced settings for the HDFS client.
		Yes	N/A

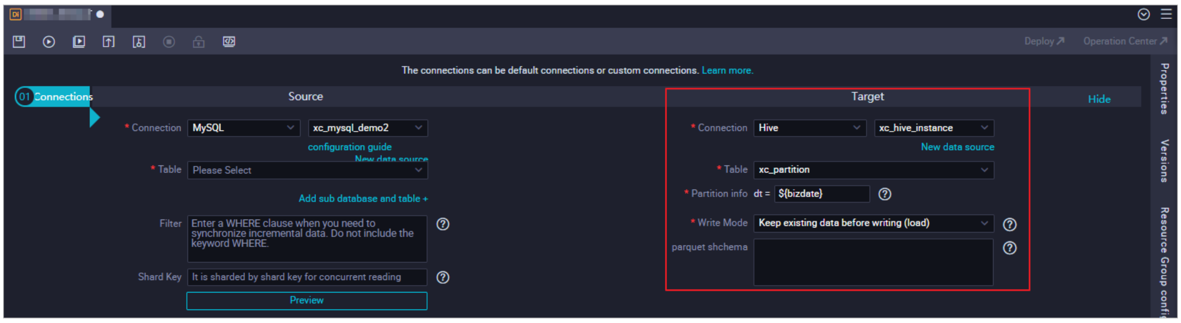
Configure Hive Writer by using the codeless UI

On the **DataStudio** page, double-click the sync node that you created. On the node configuration tab that appears, set the parameters for the node. For more information, see Configure a synchronization node by using the codeless UI.

Perform the following steps on the configuration tab of the sync node:

1. Configure the source and destination.

   Configure the source and destination data sources for the sync node.



The connections can be default connections or custom connections. Learn more.

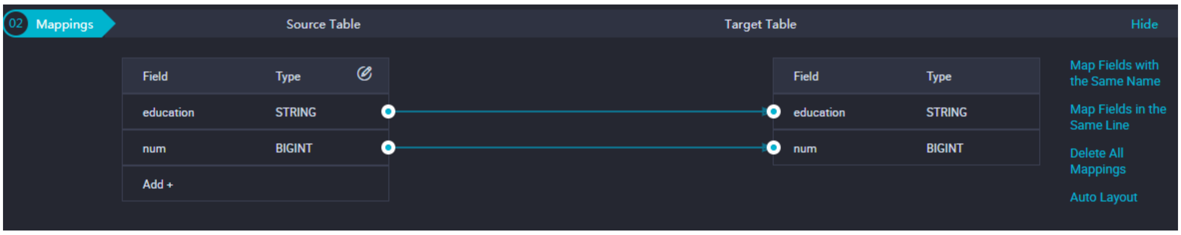
Source:
- * Connection: MySQL, xc_mysql_demo2
- * Table: Please Select
- Filter: Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.
- Shard Key: It is sharded by shard key for concurrent reading

Target:
- * Connection: Hive, xc_hive_instance
- * Table: xc_partition
- * Partition info: dt = ${bizdate}
- * Write Mode: Keep existing data before writing (load)
- parquet shchema:

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table to which you want to write data. This parameter is equivalent to the table parameter that is described in the preceding section.
Partition Key Column	The partition to which Hive Writer writes data. You must specify the lowest-level partition. Hive Writer can write data only to a single partition.
Write Mode	The write mode. This parameter is equivalent to the writeMode parameter that is described in the preceding section.

parquet shchema	The schema of the Hive table of the *parquet* type. The following example shows the configurations:
	`message tmp{`
	`REQUIRED BINARY  id;`
	`REQUIRED BINARY name;`
	`REQUIRED BINARY cyctime1;`
	`}`
	This parameter is required only if the Hive table at the underlying layer is of the *parquet* type.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.

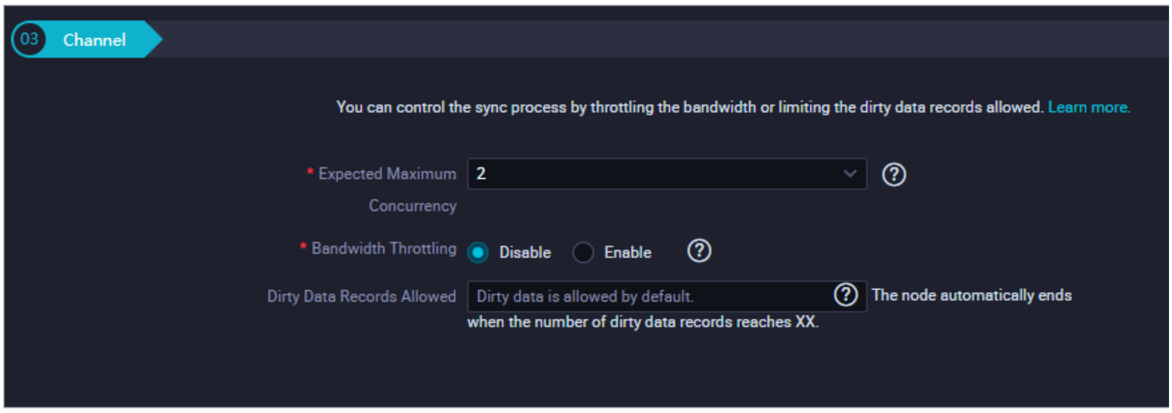


Field	Type
education	STRING
num	BIGINT

Field	Type
education	STRING
num	BIGINT

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: Dirty data is allowed by default. The node automatically ends when the number of dirty data records reaches XX.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the sync node uses to read data from the source or write data to the destination. You can configure the parallelism for the sync node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records that are allowed.

Configure Hive Writer by using the code editor

In the following code in the JSON format, a sync node is configured to write data to Hive. For more information about how to configure a sync node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type": "job",
    "steps": [
        {
            "stepType": "hive",
            "parameter": {
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "hive",
            "parameter": {
                "partition": "year=a,month=b,day=c", // The partition information of the Hive table.
                "datasource": "hive_ha_shanghai", // The name of the data source.
                "table": "partitiontable2", // The Hive table to which you want to write data.
                "column": [ // The names of the columns to which you want to write data.
                    "id",
                    "name",
                    "age"
                ],
                "writeMode": "append" // The mode in which Hive Writer loads data to the Hive table.
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": ""
        },
        "speed": {
            "throttle":true,// Specifies whether to enable bandwidth throttling. A value of false indicates tha
t bandwidth throttling is disabled, and a value of true indicates that bandwidth throttling is enabled. The mbp
s parameter takes effect only if the throttle parameter is set to true.
            "concurrent": 2, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    }
}
```

Hive versions supported by Hive Writer

5.3.31. Maxgraph Writer

This topic describes the parameters that are supported by Maxgraph Writer and how to configure Maxgraph Writer by using the codeless user interface (UI) and code editor.

Note If you want to use Maxgraph Writer to write data from MaxCompute tables to Maxgraph, grant the Maxgraph build account the read permissions on the MaxCompute tables in your MaxCompute project. Contact the Maxgraph administrator to obtain the Maxgraph build account.

To write data from MaxCompute tables to Maxgraph, you can perform the following operations:

1. Create a MapReduce job to map columns in a MaxCompute table to the vertices or edges in Maxgraph. The MapReduce job converts data records to the format that Maxgraph supports.
2. Upload the data records that are converted by the MapReduce job to the storage of Maxgraph.

Parameters

Parameter	Description	Required	Default value
endpoint	The endpoint of Maxgraph.	Yes	No default value
graphName	The name of the Maxgraph instance.	Yes	No default value
accessId	The AccessKey ID of the account that you use to connect to Maxgraph.	Yes	No default value
accessKey	The AccessKey secret of the account that you use to connect to Maxgraph.	Yes	No default value
label	The label, which is the name of the vertex or edge.	Yes	No default value
labelType	The type of the label. Valid values: vertex and edge.	Yes	No default value
srcLabel	The label of the start vertex in an edge. This parameter takes effect only when you import data about edges.	Yes	No default value
dstLabel	The label of the end vertex in an edge. This parameter takes effect only when you import data about edges.	Yes	No default value
splitSize	The size of a shard in the MapReduce job. Unit: MB.	No	*256*
onlineMode	The mode in which data is uploaded to the storage of Maxgraph. Valid values: ● partition: When data is being uploaded, both existing data records and newly uploaded data records may be queried. However, the data consistency is ensured. The data upload speed in this mode is faster than that in type mode. ● type: When data is being uploaded, only existing data records can be queried. New data records can be queried only after data is uploaded. The data upload speed in this mode is slower than that in partition mode.	No	*type*
platform			*MaxCompute*

Parameter	Description	Required	Default value
column	The name of the vertex property. This parameter takes effect only when you import data about vertices.	Yes	No default value
name	The name of the property.	Required only when you import data about edges	No default value
propertyType	The type of the property. Valid values: srcPrimaryKey, dstPrimaryKey, and edgeProperty.	Required only when you import data about edges	No default value
srcPrimaryKey	The primary key of the start vertex. This parameter takes effect only when you import data about edges.	Required only when you import data about edges	No default value
dstPrimaryKey	The primary key of the end vertex. This parameter takes effect only when you import data about edges.	Required only when you import data about edges	No default value
edgeProperty	The properties of the edge. This parameter can be left empty if the edge has no properties.	No	No default value

Configure Maxgraph Writer by using the code editor

You can configure Maxgraph Writer by using the code editor. For more information, see Create a synchronization node by using the code editor.

In the following code, synchronization nodes are configured to write data about vertices and edges to Maxgraph.

- Configure a synchronization node to write data about vertices to Maxgraph

```
{
  "job": {
    "setting": {
      "speed": {
        "channel": 1 // Set channel to 1.
      },
      "errorLimit": {
        "record": 1000
      }
    },
    "content": [
      {
        "reader": {
          "name": "odpsreader",
          "parameter": {
            "accessId": "*****",
            "accessKey": "*****",
            "project": "maxgraph_dev",
            "table": "maxgraph_demo_person",
            "column": [ // The names of columns in the MaxCompute table. The value of this parameter has a one-to-one mapping with the value of the column parameter of Maxgraph Writer.
              "id",
              "name",
              "age"
            ],
            "packageAuthorizedProject": "biggraph_dev",
            "splitMode": "record",
            "odpsServer": "******"
          }
        },
        "writer": {
          "name": "maxgraphwriter",
          "parameter": {
            "endpoint": "http://graph.alibaba.net",
            "graphName": "xxx",
            "accessId": "xxx",
            "accessKey": "xxx",
            "label": "person",
            "labelType": "vertex",
            "onlineMode": "partition",
            "platform": "odps",
            "splitSize": "256",
            "column": [ // The names of vertex properties in Maxgraph. The value of this parameter has a one-to-one mapping with the value of the column parameter of MaxCompute Reader.
              "id",
              "name",
              "age"
            ]
          }
        }
      }
    ]
  }
}
```

- Configure a synchronization node to write data about edges to Maxgraph

```
{
  "job": {
    "setting": {
      "speed": {
        "channel": 1 // Set channel to 1.
      },
      "errorLimit": {
        "record": 1000
      }
    },
    "content": [
      {
        "reader": {
          "name": "odpsreader",
          "parameter": {
            "accessId": "*****",
            "accessKey": "*****",
            "project": "maxgraph_dev",
            "table": "maxgraph_demo_knows",
            "column": [
              "person_id",
              "person_id2",
              "weight",
              "id"
            ],
            "packageAuthorizedProject": "biggraph_dev",
            "splitMode": "record",
            "odpsServer": "****"
          }
        },
        "writer": {
          "name": "maxgraphwriter",
          "parameter": {
            "endpoint": "http://graph.alibaba.net",
            "graphName": "xxx",
            "accessId": "xxx",
            "accessKey": "xxx",
            "label": "knows",
            "labelType": "edge",
            "srcLabel": "person",
            "dstLabel": "person",
            "onlineMode": "partition",
            "platform": "odps",
            "splitSize": "256",
            "column": [
              {
                "name": "id", // The name of the property in Maxgraph.
                "propertyType": "srcPrimaryKey" // The type of the property. Valid values: srcPrimaryKey, dst
PrimaryKey, and edgeProperty.
              },
              {
                "name": "id",
                "propertyType": "dstPrimaryKey"
              },
              {
                "name": "weight",
                "propertyType": "edgeProperty"
              },
              {
                "name": "id",
                "propertyType": "edgeProperty"
              }
            ]
          }
        }
      }
    ]
  }
}
```

5.3.32. Kafka Writer

Kafka Writer writes data to Kafka by using Kafka SDK for Java. This topic describes how Kafka Writer works, the parameters that are supported by Kafka Writer, and how to configure Kafka Writer by using the codeless user interface (UI) and code editor.

Notice Kafka Writer supports only exclusive resource groups for Data Integration. You cannot use default resource groups or custom resource groups.

Apache Kafka is a fast, scalable, high-throughput, and distributed messaging system that supports fault tolerance. This system is used to publish and subscribe to messages. Kafka provides built-in partitions, supports data replicas, and can be used to process a large number of messages.

How it works

Kafka Writer writes data to Kafka by using Kafka SDK for Java of the following version:

```
<dependency>
   <groupId>org.apache.kafka</groupId>
   <artifactId>kafka-clients</artifactId>
   <version>2.0.0</version>
</dependency>
```

Parameters

Parameter	Description	Required
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes
server	The address of a Kafka broker in your Kafka cluster. Specify the address in the format of IP address:Port number.	Yes
topic	The name of the Kafka topic to which you want to write data. Topics are categories in which Kafka maintains feeds of messages. Each message that is published to a Kafka cluster is assigned to a topic. Each topic contains a group of messages.	Yes
valueIndex	The sequence number of the column that is obtained from a reader and used as the value in the Kafka topic. If you leave this parameter empty, all columns obtained from the reader are concatenated by using the delimiter specified by the fieldDelimiter parameter to form the value.	No

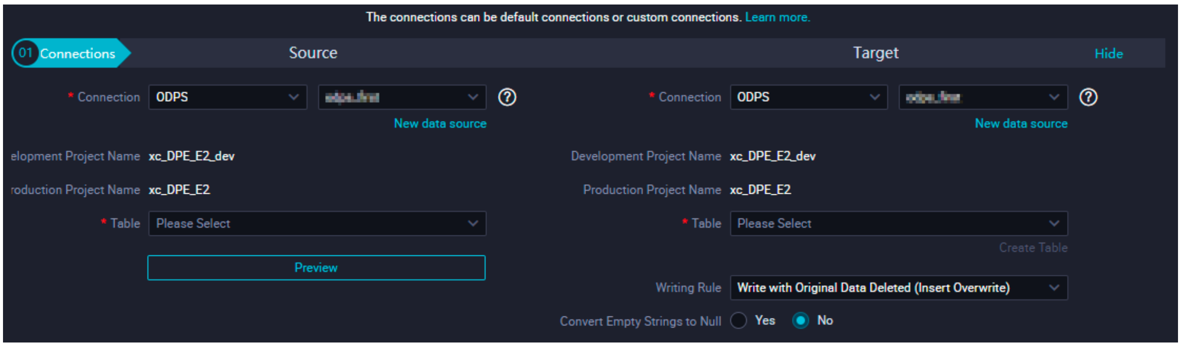
Parameter	Description	Required
writeMode	The write mode. If you leave the valueIndex parameter empty, you can use the writeMode parameter to specify the format in which Kafka Writer concatenates all columns obtained from the reader. The default value is text. Valid values: • text: Kafka Writer uses the delimiter specified by the fieldDelimiter parameter to concatenate all columns obtained from the reader. • JSON: Kafka Writer concatenates all columns obtained from the reader into a JSON string based on the column names specified by the column parameter. For example, three columns are obtained from the reader, and the values in the three columns are a, b, and c. If you set the writeMode parameter to text and the fieldDelimiter parameter to a number sign (#), Kafka Writer concatenates the columns into the string a#b#c and writes this string to Kafka. If you set the writeMode parameter to JSON and the column parameter to [{"name":"col1"},{"name":"col2"},{"name":"col3"}], Kafka Writer concatenates the columns into the JSON string {"col1":"a","col2":"b","col3":"c"} and writes this JSON string to Kafka. If you specify the valueIndex parameter, the writeMode parameter is invalid.	No
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as "column":["id","name","age"]. If you want to write data to all the columns in the destination table, set this parameter to an asterisk (*), such as "column":["*"]. If you leave the valueIndex parameter empty and set the writeMode parameter to JSON, the column parameter determines the names of the columns in the JSON string into which the columns obtained from the reader are concatenated, such as "column":[{"name":id"},{"name":"name"},{"name":"age"}]. • If the number of columns that are obtained from the reader is greater than the number of columns that are specified in the column parameter, Kafka Writer skips the excessive columns. Example: Three columns are obtained from the reader, and the values in the columns are a, b, and c. If the column parameter is set to [{"name":"col1"},{"name":"col2"}], Kafka Writer concatenates the columns into the JSON string {"col1":"a","col2":"b"} and writes this JSON string to Kafka. • If the number of columns that are obtained from the reader is less than the number of columns that are specified in the column parameter, Kafka Writer sets the values of the excess columns in Kafka to null or the string that is specified by the nullValueFormat parameter. Example: Two columns are obtained from the reader, and the values in the columns are a and b. If the column parameter is set to [{"name":"col1"},{"name":"col2"},{"name":"col3"}], Kafka Writer concatenates the columns into the JSON string {"col1":"a","col2":"b","col3":null} and writes this JSON string to Kafka. If you specify the valueIndex parameter or set the writeMode parameter to text, the column parameter is invalid.	Required if the valueIndex parameter is not specified and the writeMode parameter is set to JSON
partition	The ID of the partition to which you want to write data. The value of this parameter must be an integer that is greater than or equal to 0.	No

Parameter	Description	Required
keyIndex	The sequence number of the column that is obtained from the reader and used as the key in the Kafka topic. The value of this parameter must be an integer that is greater than or equal to 0. If you set this parameter to an integer less than 0, an error occurs.	No
keyIndexes	The sequence numbers of the columns that are obtained from the reader and used as the key in the Kafka topic. The sequence numbers must start from 0 and be separated by commas (,), such as [0,1,2]. If you leave this parameter empty, the key is null, and data is written to each partition in the Kafka topic in turn. You can set only one of the keyIndex and keyIndexes parameters.	No
fieldDelimiter	The column delimiter. If you set the writeMode parameter to text and leave the valueIndex parameter empty, Kafka Writer uses the specified column delimiter to concatenate all columns that are obtained from the reader to form the value. You can use a single character or multiple characters as the column delimiter. The characters can be Unicode characters such as *\u0001*, or escape characters such as *\t* or *\n*. Default value: *\t*. If the writeMode parameter is not set to text or the valueIndex parameter is specified, the fieldDelimiter parameter is invalid.	No
keyType	The data type of the key in the Kafka topic. Valid values: BYTEARRAY, DOUBLE, FLOAT, INTEGER, LONG, and SHORT.	Yes
valueType	The data type of the value in the Kafka topic. Valid values: BYTEARRAY, DOUBLE, FLOAT, INTEGER, LONG, and SHORT.	Yes
nullKeyFormat	If the column specified in the keyIndex or keyIndexes parameter contains the value null, Kafka Writer replaces null with the value of the nullKeyFormat parameter. If you leave the nullKeyFormat parameter empty, Kafka Writer retains the value null.	No
nullValueFormat	If a column obtained from the reader contains the value null, Kafka Writer replaces null with the value of the nullValueFormat parameter. If you leave the nullValueFormat parameter empty, Kafka Writer retains the value null.	No
acks	The acknowledgment configuration used when the Kafka producer is initialized. This parameter specifies the method used to confirm that data is written to Kafka. Default value: all. Valid values: • 0: A Kafka producer does not acknowledge whether data is written to the destination. • 1: A Kafka producer acknowledges that the write operation is successful if data is written to the primary replica. • all: A Kafka producer acknowledges that the write operation is successful if data is written to all replicas.	No

Configure Kafka Writer by using the codeless UI

1. Configure data sources.

Set parameters in the **Source** and **Target** sections for the synchronization node.



The connections can be default connections or custom connections. Learn more.

01 Connections — Source — Target — Hide

Source:
- * Connection: ODPS, [redacted] — New data source
- elopment Project Name: xc_DPE_E2_dev
- roduction Project Name: xc_DPE_E2
- * Table: Please Select
- Preview

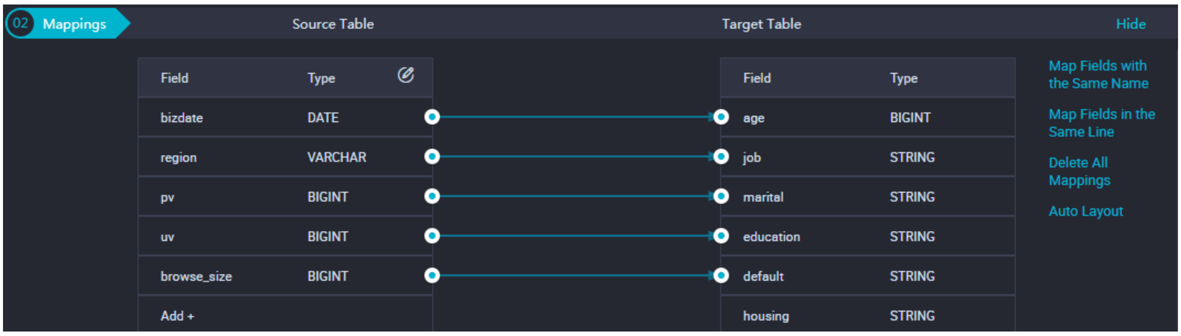
Target:
- * Connection: ODPS, [redacted] — New data source
- Development Project Name: xc_DPE_E2_dev
- Production Project Name: xc_DPE_E2
- * Table: Please Select — Create Table
- Writing Rule: Write with Original Data Deleted (Insert Overwrite)
- Convert Empty Strings to Null: ○ Yes ● No

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter described in the preceding section.
Topic	The name of the Kafka topic to which you want to write data. This parameter is equivalent to the topic parameter described in the preceding section.
Key Columns	The sequence numbers of the columns that are obtained from the reader and used as the key in the Kafka topic. This parameter is equivalent to the keyIndexes parameter described in the preceding section.
Write Mode	The write mode. This parameter is equivalent to the writeMode parameter described in the preceding section.
Delimiter	The column delimiter. This parameter is equivalent to the fieldDelimiter parameter described in the preceding section. If you set the Write Mode parameter to text, Kafka Writer uses the specified column delimiter to concatenate all columns that are obtained from the reader to form the value.
Substitute For Null Key	This parameter is equivalent to the nullKeyFormat parameter described in the preceding section.
Substitute For Null Value	This parameter is equivalent to the nullValueFormat parameter described in the preceding section.
acks	The acknowledgment configuration used when the Kafka producer is initialized. This parameter is equivalent to the acks parameter described in the preceding section.
Batch Config	Determines the values of the batch.size and linger.ms parameters used when the Kafka producer is initialized. The default values of the two parameters are 16384 and 10. You can set this parameter to limit the amount of data to be written at a time.
Timeout	Determines the values of the timeout.ms, request.timeout.ms, and metadata.fetch.timeout.ms parameters used when the Kafka producer is initialized. The default values of the three parameters are 30000, 30000, and 60000. You can set this parameter to limit the timeout period when the data is written at a time.

2. Configure column mappings. This operation is equivalent to setting the column parameter when you use the code editor. Columns in the source on the left have a one-to-one mapping with columns in the destination on the right.

   - If you set the Write Mode parameter to text, the specified column delimiter is used to concatenate all columns that are obtained from the reader to form the value.
   - If you set the Write Mode parameter to JSON, the names of columns concatenated into the JSON string in the destination are the names of columns that are read from the source. For example, two columns are obtained from the source, and the values in the columns are a and b. If the column names are col1 and col2, Kafka Writer concatenates the columns into the JSON string {"col1":"a","col2":"b"} and writes this JSON string to Kafka.

   Note The names of columns in the destination must contain letters, digits, or underscores (_). Otherwise, Kafka Writer fails to write data.



Field	Type	Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
		housing	STRING

GUI element	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between columns with the same name. The data types of the columns must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between columns in the same row. The data types of the columns must match.
Delete All Mappings	Click **Delete All Mappings** to remove the established mappings.
Auto Layout	Click Auto Layout to sort the columns based on specific rules.

3.

Configure Kafka Writer by using the code editor

You can configure Kafka Writer by using the code editor. For more information, see Create a synchronization node by using the code editor.

The following code shows a configuration example:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"Kafka",// The type of the writer.
            "parameter":{
                   "server": "ip:9092", // The address of a Kafka broker.
                   "keyIndex": 0, // The column that is used as the key. You must use the lower camel case for
the column name.
                   "valueIndex": 1, // The column that is used as the value. You can specify only one column. I
f you leave this parameter empty, all columns obtained from the reader are used as the value.
        // If you want to use the second, third, and fourth columns in a MaxCompute table as the value, cleanse
and integrate the data in the table. Then, create a MaxCompute table, write the processed data to the new table
, and then use the new table to synchronize data.
                   "keyType": "Integer", // The data type of the key in the Kafka topic.
                   "valueType": "Short", // The data type of the value in the Kafka topic.
                   "topic": "t08", // The name of the Kafka topic to which you want to write data.
                   "batchSize": 1024 // The number of data records to write at a time.
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
                     "throttle":true,// Specifies whether to enable bandwidth throttling. A value of false indi
cates that bandwidth throttling is disabled, and a value of true indicates that bandwidth throttling is enabled
. The mbps parameter takes effect only when the throttle parameter is set to true.
                     "concurrent":1, // The maximum number of parallel threads.
                     "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

Use SASL authentication

If you want to use SASL or SSL for authentication, configure the SASL or SSL authentication mode when you configure a Kafka data source. For more information, see Add a Kafka data source.

5.3.33. Vertica Writer

Vertica is a column-oriented database that uses the massively parallel processing (MPP) architecture. Vertica Writer writes data to tables that are stored in Vertica databases. This topic describes the working principles and parameters that are supported by Vertica Writer and how to configure Vertica Writer by using the codeless user interface (UI) and code editor.

Notice Vertica Writer supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.

How it works

Vertica Writer connects to a remote Vertica database by using Java Database Connectivity (JDBC) and executes the `INSERT INTO` statement to write data to the Vertica database. Data is submitted to the Vertica database in batches.

Vertica Writer is designed for extract, transform, load (ETL) developers to import data in data warehouses to Vertica databases. Vertica Writer can also be used as a data migration tool by users such as database administrators.

Vertica Writer obtains data from a reader and generates an SQL statement based on your configurations.

- `INSERT INTO`: If a primary key conflict or unique index conflict occurs, data cannot be written to the conflicting rows.
- Data can be written to tables that are stored only in primary Vertica databases.

  Note A synchronization node that uses Vertica Writer must have at least the permissions to execute the `INSERT INTO` statement. Whether other permissions are required depends on the SQL statements that you specify in the preSql and postSql parameters when you configure the node.

- Vertica Writer does not support the writeMode parameter.
- Vertica Writer connects to a Vertica database by using a Vertica JDBC driver. Make sure that the driver version is compatible with your Vertica database. Vertica Writer uses the Vertica JDBC driver of the following version:

```
<dependency>
    <groupId>com.vertica</groupId>
    <artifactId>vertica-jdbc</artifactId>
    <version>7.1.2</version>
</dependency>
```

Parameters

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
jdbcUrl	The JDBC URL of the Vertica database. The jdbcUrl parameter must be included in the connection parameter. • You can configure only one JDBC URL for a database. Vertica Writer cannot write data to a database that has multiple primary databases. • The format of the value specified for this parameter must comply with the official specifications of Vertica. You can also specify the information of the attachment facility. Example: `jdbc:vertica://127.0.0.1:3306/database`.	Yes	No default value
username	The username that you use to connect to the database.	Yes	No default value
password	The password that you use to connect to the database.	Yes	No default value
table	The name of the table to which you want to write data. Specify the name in a JSON array. **Note** The table parameter must be included in the connection parameter.	Yes	No default value

Parameter	Description	Required	Default value
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as `"column": ["id", "name", "age"]` .	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. Use `@table` to indicate the name of the destination table in the SQL statement. When you execute this SQL statement, @table is replaced by the name of the destination table.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run.	No	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and Vertica and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

Configure Vertica Writer by using the codeless UI

This method is not supported.

Configure Vertica Writer by using the code editor

In the following code, a synchronization node is configured to write data to a Vertica database. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"vertica",// The writer type.
            "parameter":{
                "datasource": "The name of the data source",
                "username": "",
                "password": "",
                "column": [// The names of the columns to which you want to write data.
                    "id",
                    "name"
                ],
                "connection": [
                    {
                        "table": [// The name of the table to which you want to write data.
                            "vertica_table"
                        ],
                        "jdbcUrl": "jdbc:vertica://ip:port/database"
                    }
                ],
                "preSql": [ // The SQL statement that you want to execute before the synchronization node is ru
n.
                    "delete from @table where db_id = -1"
                ],
                "postSql": [// The SQL statement that you want to execute after the synchronization node is run
.
                    "update @table set db_modify_time = now() where db_id = 1"
                ]
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
                    "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indica
tes that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. T
he mbps parameter takes effect only when the throttle parameter is set to true.
                    "concurrent":1, // The maximum number of parallel threads.
                    "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.34. GBase8a Writer

This topic describes the parameters that are supported by GBase8a Writer and how to configure GBase8a Writer by using the codeless user interface (UI) and code editor.

GBase 8a is a new type of column-oriented analytical database. GBase8a Writer can write data to GBase 8a databases.

> **Notice** GBase8a Writer supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.

GBase8a Writer connects to a remote GBase 8a database by using Java Database Connectivity (JDBC) and executes an `INSERT INTO` statement to write data to the GBase 8a database. The data is written to the database in batches.

GBase8a Writer is designed for extract, transform, load (ETL) developers to import data in data warehouses to GBase 8a databases. GBase8a Writer can also be used as a data migration tool by users such as database administrators.

GBase8a Writer obtains data from a reader and generates an INSERT INTO statement based on your configurations.

Limits

- `INSERT INTO`: If a primary key conflict or unique index conflict occurs, data cannot be written to conflicting rows.
- Data can be written to tables stored only in the primary database.

  > **Note** A synchronization node that uses GBase8a Writer must have at least the permissions to execute `INSERT INTO` statements. Whether other permissions are required depends on the SQL statements that you specify in the preSql and postSql parameters when you configure the node.

- GBase8a Writer does not support the writeMode parameter.
- GBase8a Writer connects to a GBase 8a database by using the MySQL database driver. You must make sure that the GBase 8a database is compatible with the driver version. GBase8a Writer uses the MySQL database driver of the following version:

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.22</version>
</dependency>
```

Parameters

Parameter	Description	Required	Default value
jdbcUrl	The JDBC URL of the GBase 8a database. This parameter is included in the connection parameter. • You can configure only one JDBC URL for a database. GBase8a Writer cannot write data to a database that has multiple primary databases. • The value format of the jdbcUrl parameter must follow official GBase 8a specifications. You can also specify the information of the attachment facility. Example: `jdbc:mysql://127.0.0.1:3306/database`.	Yes	No default value
username	The username that is used to connect to the database.	Yes	No default value
password	The password that is used to connect to the database.	Yes	No default value

Parameter	Description	Required	Default value
table	The name of the table to which you want to write data. Specify the name in a JSON array. ⓘ **Note** The table parameter must be included in the connection parameter.	Yes	No default value
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as `"column": ["id", "name", "age"]` . ⓘ **Note** The column parameter cannot be left empty.	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. Use `@table` to indicate the name of the destination table in the SQL statement. When you execute this SQL statement, @table is replaced by the name of the destination table.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run.	No	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and GBase 8a and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

Configure GBase8a Writer by using the codeless UI

This method is not supported.

Configure GBase8a Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to write data to a GBase 8a database:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"gbase8a",// The writer type.
            "parameter":{
                "datasource": "The name of the data source",
                "username": "",
                "password": "",
                "column": [// The names of the columns to which you want to write data.
                    "id",
                    "name"
                ],
                "connection": [
                    {
                        "table": [// The name of the table to which you want to write data.
                            "Gbase8a_table"
                        ],
                        "jdbcUrl": "jdbc:mysql://ip:port/database"
                    }
                ],
                "preSql": [ // The SQL statement that you want to execute before the synchronization node is run.
                    "delete from @table where db_id = -1"
                ],
                "postSql": [// The SQL statement that you want to execute after the synchronization node is run.
                    "update @table set db_modify_time = now() where db_id = 1"
                ]
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.35. ClickHouse Writer

ClickHouse is an open source column-oriented database management system (DBMS) for online analytical processing (OLAP) of queries. This topic describes how ClickHouse Writer works, the parameters that are supported by ClickHouse Writer, and how to configure ClickHouse Writer by using the codeless user interface (UI) and code editor.

Limits

- Only Alibaba Cloud ApsaraDB for ClickHouse is supported.
- ClickHouse Writer supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration, Use the default resource group, and Create a custom resource group for Data Integration.
- ClickHouse Writer connects to a ClickHouse database by using Java Database Connectivity (JDBC) and can write data to a destination table in the ClickHouse database only by using JDBC Statement.
- ClickHouse Writer allows you to specify the columns to which you want to write data. You can specify the columns in an order different from the order specified by the schema of the destination table.
- If ClickHouse Writer writes data in INSERT mode, we recommend that you throttle the transactions per second (TPS) to 1,000 to prevent high workloads on ClickHouse.
- After ClickHouse Writer writes all required data, ClickHouse Writer performs a single-process POST Flush operation to update the data records in the ClickHouse database.
- You must make sure that the driver version is compatible with your ClickHouse database. ClickHouse Writer supports only the following version of the ClickHouse database driver:

```
<dependency>
    <groupId>ru.yandex.clickhouse</groupId>
    <artifactId>clickhouse-jdbc</artifactId>
    <version>0.2.4.ali2-SNAPSHOT</version>
</dependency>
```

Background information

ClickHouse Writer writes data to ClickHouse databases. ClickHouse Writer connects to a remote ClickHouse database by using JDBC and executes the `INSERT INTO` statement to write data to the ClickHouse database.

ClickHouse Writer is designed for extract, transform, load (ETL) developers to import data in data warehouses to ClickHouse databases. ClickHouse Writer can also be used as a data migration tool by users such as database administrators.

ClickHouse Writer obtains data from a reader, generates an INSERT INTO statement based on your configurations, and executes the INSERT INTO statement to write data to ClickHouse databases.

Parameters

Parameter	Description	Required	Default value
jdbcUrl	The JDBC URL of the ClickHouse database. This parameter is included in the connection parameter. • You can configure only one JDBC URL for a database. • The value format of the jdbcUrl parameter must be in accordance with the official specifications of ClickHouse. You can also specify additional JDBC connection properties in the value of this parameter. Example: `jdbc:clickhouse://127.0.0.1:3306/database`.	Yes	No default value
username	The username that you can use to connect to the database.	Yes	No default value
password	The password that you can use to connect to the database.	Yes	No default value

Parameter	Description	Required	Default value
table	The name of the table to which you want to write data. Specify the name in a JSON array. **Note** The table parameter must be included in the connection parameter.	Yes	No default value
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as `"column": ["id", "name", "age"]` . **Note** The column parameter cannot be left empty.	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run.	No	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and ClickHouse and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

Configure ClickHouse Writer by using the codeless UI

This method is not supported.

Configure ClickHouse Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

Note Delete the comments from the following code before you run the code.

In the following code, a synchronization node is configured to write data to a ClickHouse database:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"clickhouse",// The writer type.
            "parameter":{
                "username": "",
                "password": "",
                "column": [// The names of the columns to which you want to write data.
                    "id",
                    "name"
                ],
                "connection": [
                    {
                        "table": [// The name of the table to which you want to write data.
                            "ClickHouse_table"
                        ],
                        "jdbcUrl": "jdbc:clickhouse://ip:port/database"
                    }
                ],
                "preSql": [ // The SQL statement that you want to execute before the synchronization node is ru
n.
                    "delete from table where db_id = -1"
                ],
                "postSql": [// The SQL statement that you want to execute after the synchronization node is run
.
                    "update table set db_modify_time = now() where db_id = 1"
                ],
                "batchSize": "1024",
                "batchByteSize": "67108864",
                "writeMode": "insert"
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.36. ApsaraDB for OceanBase Writer

ApsaraDB for OceanBase is a financial-grade distributed relational database that is developed by Alibaba Cloud and Ant Financial. This topic describes the parameters that are supported by ApsaraDB for OceanBase Writer and how to configure ApsaraDB for OceanBase Writer by using the codeless user interface (UI) and code editor.

> **Notice** ApsaraDB for OceanBase Writer supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.

Background information

ApsaraDB for OceanBase supports automated and non-disruptive disaster recovery across cities based on the Five Data Centers Across Three Regions solution. ApsaraDB for OceanBase provides high availability for financial services based on conventional hardware. ApsaraDB for OceanBase is a database service developed by Alibaba Cloud. It provides the online scaling capability and has undergone strict verification in terms of functionality, stability, scalability, and performance.

ApsaraDB for OceanBase Writer is designed for extract, transform, load (ETL) developers to import data in data warehouses to ApsaraDB for OceanBase databases. ApsaraDB for OceanBase Writer can also be used as a data migration tool by users such as database administrators.

ApsaraDB for OceanBase Writer obtains data from a reader and generates an SQL statement based on your configurations.

Limits

- `insert into`: If no primary key conflict or unique index conflict occurs, data is directly written to the destination table. If a primary key conflict or unique index conflict occurs, data cannot be written to the conflicting rows. In Oracle tenant mode, ApsaraDB for OceanBase supports only the `insert into` write mode.
- `insert into...ON DUPLICATE KEY UPDATE`: If no primary key conflict or unique index conflict occurs, data is processed in the same way as that in `insert into` mode. If a conflict occurs, the specified fields in the rows in the destination table are updated. In MySQL tenant mode, ApsaraDB for OceanBase supports both the `insert into` and `insert into...ON DUPLICATE KEY UPDATE` write modes.
- ApsaraDB for OceanBase Writer can write data to tables only in the primary ApsaraDB for OceanBase database.

  > **Note** A synchronization node that uses ApsaraDB for OceanBase Writer must have at least the permissions to execute the `INSERT INTO` statement. Whether other permissions are required depends on the SQL statements that you specify in the preSql and postSql parameters when you configure the node.

- We recommend that you write data to the destination table in batches. ApsaraDB for OceanBase Writer submits a write request when the number of rows reaches a specific threshold.
- ApsaraDB for OceanBase supports the Oracle and MySQL tenant modes. Make sure that the SQL statements that you specify in the preSql and postSql parameters comply with the related SQL syntax. Otherwise, the SQL statements may fail to be executed.
- ApsaraDB for OceanBase Writer connects to an ApsaraDB for OceanBase database by using an OceanBase database driver. Make sure that the driver version is compatible with your ApsaraDB for OceanBase database. ApsaraDB for OceanBase Writer uses the OceanBase database driver of the following version:

```
<dependency>
    <groupId>com.alipay.OceanBase</groupId>
    <artifactId>OceanBase-connector-java</artifactId>
    <version>3.1.0</version>
</dependency>
```

Parameters

Parameter	Description	Required	Default value

Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can connect to the ApsaraDB for OceanBase database based on the setting of the jdbcUrl or username parameter.	No	No default value
jdbcUrl	The Java Database Connectivity (JDBC) URL of the ApsaraDB for OceanBase database. This parameter is included in the connection parameter. - You can configure only one JDBC URL for a database. ApsaraDB for OceanBase Writer cannot write data to a database that has multiple primary databases. - The format of the value specified for this parameter must comply with the official specifications of ApsaraDB for OceanBase. You can also specify the information of the attachment facility. Example: `jdbc:mysql://127.0.0.1:3306/database` .	Yes	No default value
username	The username that you use to connect to the database.	Yes	No default value
password	The password that you use to connect to the database.	Yes	No default value
table	The name of the table to which you want to write data. Specify the name in a JSON array. **Note** The table parameter must be included in the connection parameter.	Yes	No default value
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as `"column": ["id", "name", "age"]` . **Note** The column parameter cannot be left empty.	Yes	No default value
writeMode	The write mode. Valid values: `insert into` and `insert into...ON DUPLICATE KEY UPDATE` .	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. Use `@table` to indicate the name of the destination table in the SQL statement. When you execute this SQL statement, @table is replaced by the name of the destination table.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run.	No	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and the database and increases throughput. **Note** If you set this parameter to a value greater than 2048, an out of memory (OOM) error may occur during data synchronization.	No	*1,024*

Configure ApsaraDB for OceanBase Writer by using the codeless UI

This method is not supported.

Configure ApsaraDB for OceanBase Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to write data to an ApsaraDB for OceanBase database:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"apsaradb_for_OceanBase",// The writer type.
            "parameter":{
                "datasource": "The name of the data source",
                "column": [// The names of the columns to which you want to write data.
                    "id",
                    "name"
                ],
                "table": "apsaradb_for_OceanBase_table",// The name of the table to which you want to write dat
a.
                "preSql": [ // The SQL statement that you want to execute before the synchronization node is ru
n.
                    "delete from @table where db_id = -1"
                ],
                "postSql": [// The SQL statement that you want to execute after the synchronization node is run
.
                    "update @table set db_modify_time = now() where db_id = 1"
                ],
                "writeMode": "insert",
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.37. Hologres Writer

Hologres Writer writes data to Hologres and helps you analyze the data in real time.

> **Notice** Hologres Writer supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration, Use the default resource group, and Create a custom resource group for Data Integration.

Limits

Hologres Writer cannot write data to the external tables of Hologres.

How it works

Hologres Writer obtains data from a reader and writes data to Hologres based on the values of the writeMode and conflictMode parameters.

- If you set the writeMode parameter to sdk, Hologres Writer writes data to Hologres by using the HoloHub API. This write mode can help implement optimal data synchronization performance.
- If you set the writeMode parameter to insert, Hologres Writer writes data to Hologres by using Java Database Connectivity (JDBC) to execute the `INSERT INTO` statement that is provided by PostgreSQL. We recommend that you use this write mode.

You can use the conflictMode parameter to specify how to process conflicting data if a primary key conflict occurs.

- If you set the conflictMode parameter to replace, new data overwrites existing data.
- If you set the conflictMode parameter to ignore, existing data is retained, and new data is ignored.

In different write modes, different methods are used to process the conflicting data. If you set the writeMode parameter to sdk, you can configure the properties of the destination Hologres table to change the method that is used to process the conflicting data.

> **Notice** The conflictMode parameter is suitable only for tables that have primary keys.

Parameters

Parameter	Description	Required	Default value

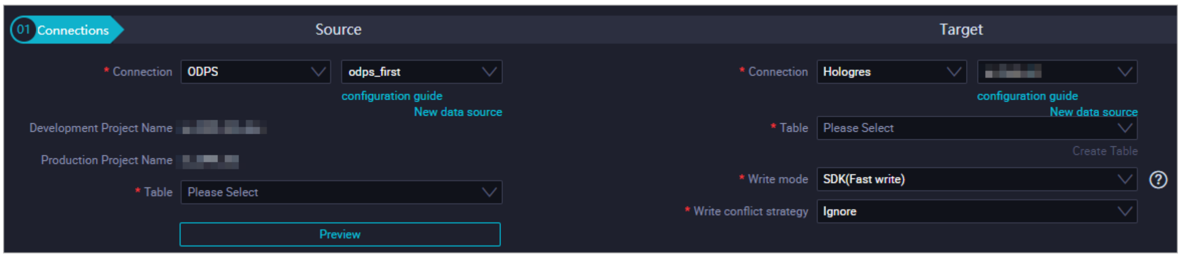
Parameter	Description	Required	Default value
endpoint	The endpoint of the destination Hologres instance. Specify the value in the format of `instance-id-region-endpoint.hologres.aliyuncs.com:Port number`. You can view the endpoint of a Hologres instance on the configuration page of the instance in the Hologres console. The endpoint of a Hologres instance varies based on the network type. Network types include the classic network, Internet, and virtual private cloud (VPC). Set this parameter based on the type of the network where your exclusive resource group for Data Integration and the Hologres instance reside. If an invalid endpoint is specified, the connection between the exclusive resource group for Data Integration and the Hologres instance may fail, or data synchronization performance may be poor. The endpoints for the three network types are in the following formats: • Public endpoint: `instance-id-region-endpoint.hologres.aliyuncs.com:Port number` • Classic network endpoint: `instance-id-region-endpoint-internal.hologres.aliyuncs.com:Port number` • VPC endpoint: `instance-id-region-endpoint-vpc.hologres.aliyuncs.com:Port number` We recommend that you deploy the exclusive resource group for Data Integration and the Hologres instance in the same zone of the same region. This helps ensure a successful network connection and optimal data synchronization performance.	Yes	No default value
accessId	The AccessKey ID of the account that you use to connect to Hologres.	Yes	No default value
accessKey	The AccessKey secret of the account that you use to connect to Hologres. Make sure that the account is authorized to write data to the destination table.	Yes	No default value
database	The name of the destination database in the Hologres instance.	Yes	No default value
table	The name of the destination Hologres table. You can specify the table name in the format of `Schema name.Table name`.	Yes	No default value
writeMode	The write mode. Valid values: sdk and insert. For more information about the valid values, see How it works. In the code editor, you can set the following parameters if you use the sdk mode: • maxCommitSize: the maximum size of data that Hologres Writer can write to Hologres at a time. Unit: bytes. Default value: *1048576*. This parameter is optional. • maxRetryCount: the maximum number of retries allowed if a data write error occurs. Default value: *500*. This parameter is optional. • retryInterval: the interval at which Hologres Writer performs retries. Unit: milliseconds. Default value: 1000. This parameter is optional.	Yes	No default value
conflictMode	The mode in which the conflicting data is processed. Valid values: replace and ignore. For more information about the valid values, see How it works.	Yes	No default value

Parameter	Description	Required	Default value
column	The names of the columns to which you want to write data. The names of the primary key columns in the destination table must be included. If you want to write data to all the columns in the destination table, set this parameter to `["*"]`.	Yes	No default value
partition	The destination partition. Specify the partition in the format of `partition key column=value`. This parameter is valid only for partitioned tables. Note • Hologres supports only list partitioning, and you can specify only one column as the partition key column. The data type of the partition key column must be INT4 or TEXT. • The value of this parameter must match the partition configuration in the data definition language (DDL) statements that are used to create the Hologres table.	No	*Left empty, which indicates that the destination table is a non-partitioned table*

Configure Hologres Writer by using the codeless UI

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.



01 Connections

Source

* Connection: ODPS, odps_first

Development Project Name: [redacted]

Production Project Name: [redacted]

* Table: Please Select

Target

* Connection: Hologres, [redacted]

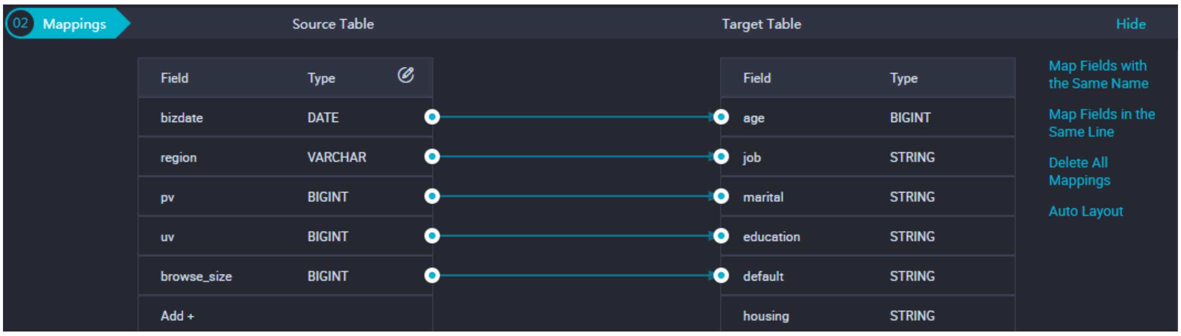
* Table: Please Select

* Write mode: SDK(Fast write)

* Write conflict strategy: Ignore

Parameter	Description
Connection	The name of the data source to which you want to write data.
Table	The name of the table to which you want to write data. This parameter is equivalent to the table parameter that is described in the preceding section.
Write Mode	The write mode. This parameter is equivalent to the writeMode parameter that is described in the preceding section.
Write conflict strategy	The write conflict strategy. This parameter is equivalent to the conflictMode parameter that is described in the preceding section.

2. Configure field mapping. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.



02 Mappings

Source Table

Field	Type
bizdate	DATE
region	VARCHAR
pv	BIGINT
uv	BIGINT
browse_size	BIGINT
Add +	

Target Table

Field	Type
age	BIGINT
job	STRING
marital	STRING
education	STRING
default	STRING
housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3.

Configure Hologres Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor. The following sample code provides examples on how to configure a synchronization node to write data to a non-partitioned table and how to configure a synchronization node to write data to a partitioned table.

- Write data to a non-partitioned table

- In the following code, a synchronization node is configured to write the data obtained from a reader to a non-partitioned Hologres table in sdk mode:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"holo",
            "parameter":{
              "endpoint": "instance-id-region-endpoint.hologres.aliyuncs.com:port",
                "accessId": "<yourAccessKeyId>", // The AccessKey ID of the account that you use to connec
t to Hologres.
                "accessKey": "<yourAccessKeySecret>", // The AccessKey secret of the account that you use
to connect to Hologres.
                "database": "postgres",
                "table": "<yourTableName>",
                "writeMode": "sdk",
                "conflictMode": "replace",
                "column" : [
                    "tag",
                    "id",
                    "title"
                ],
                "maxCommitSize": 1048576,
                "maxRetryCount": 500
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty records allowed.
        },
        "speed":{
                "throttle": true,// Specifies whether to enable bandwidth throttling. The value false indi
cates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is ena
bled. The mbps parameter takes effect only when the throttle parameter is set to true.
                "concurrent":1, // The maximum number of parallel threads.
                "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

  - The following sample DDL statements are used to create a non-partitioned Hologres table:

    ```
    begin;
    drop table if exists test_holowriter_sdk_replace;
    create table test_holowriter_sdk_replace(
      tag text not null,
      id int not null,
      body text not null
      primary key (tag, id));
      call set_table_property('test_holowriter_sdk_replace', 'orientation', 'column');
      call set_table_property('test_holowriter_sdk_replace', 'shard_count', '3');
    commit;
    ```

- Write data to a partition in a partitioned table

- In the following code, a synchronization node is configured to write the data obtained from a reader to a partition in a partitioned Hologres table in sdk mode:

  Note Exercise caution when you set the partition parameter.

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"holo",
            "parameter":{
              "endpoint": "instance-id-region-endpoint.hologres.aliyuncs.com:port",
                "accessId": "<yourAccessKeyId>", // The AccessKey ID of the account that you use to connec
t to Hologres.
                "accessKey": "<yourAccessKeySecret>", // The AccessKey secret of the account that you use
to connect to Hologres.
                "database": "postgres",
                "table": "<yourTableName>",
                "writeMode": "sdk",
                "conflictMode": "ignore",
                "column" : [
                    "*"
                ],
                "partition": "tag=foo",
                "maxCommitSize": 1048576,
                "maxRetryCount": 500
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"// The maximum number of dirty records allowed.
        },
        "speed":{
            "throttle": true,// Specifies whether to enable bandwidth throttling. The value false indicate
s that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled
. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

- The following sample DDL statements are used to create a partitioned Hologres table:

```
begin;
drop table if exists test_holowriter_part_table_sdk_ignore;
create table test_holowriter_part_table_sdk_ignore(
  tag text not null,
  id int not null,
  title text not null,
  body text,
  primary key (tag, id))
  partition by list( tag );
  call set_table_property('test_holowriter_part_table_sdk_ignore', 'orientation', 'column');
  call set_table_property('test_holowriter_part_table_sdk_ignore', 'shard_count', '3');
commit;
```

5.3.38. RestAPI Writer

This topic describes the parameters that are supported by RestAPI Writer and how to configure RestAPI Writer by using the codeless user interface (UI) and code editor.

Context

RestAPI Writer can send requests to RESTful APIs to write data. RestAPI Writer receives data from a reader, generates JSON-formatted data based on the setting of the column parameter, and then sends requests to RESTful APIs to transfer the data.

Parameters

To implement data integration, you must add a data source and configure it as the source or destination. You must also configure the data that you want to integrate and the data types. During data integration, a reader extracts data from the source, and a writer writes data to the destination.

The following table describes the parameters that you must configure when you use RestAPI Writer to write data to a RestAPI data source.

Note You must configure the parameters that are described in the following table when you add a RestAPI data source and configure a data synchronization node.

Scheduling parameters are not supported for data synchronization nodes that use RestAPI Writer.

Parameter	Description	Required	Default value
url	The URL of the RESTful API.	Yes	No default value
dataMode	The format in which RESTful Writer transfers JSON-formatted data. - oneData: RestAPI Writer transfers one data record in each request. - multiData: RestAPI Writer transfers multiple data records in each request. The number of requests is determined by the number of tasks generated by the reader.	Yes	No default value

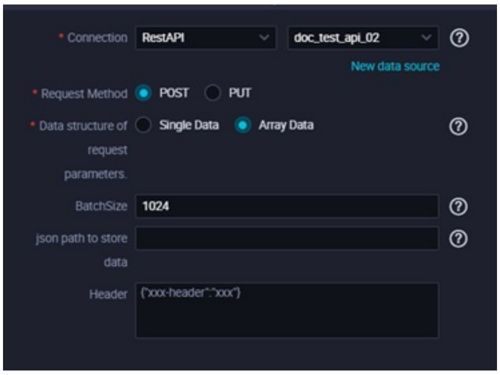
Parameter	Description	Required	Default value
column	The columns to which you want to write the generated JSON-formatted data. The type field specifies the data type of a column. The name field specifies the JSON-formatted path where the column is stored. You can also specify the column parameter in the following format: "column":[{"type":"long","name":"a.b" // Store data in the a.b path.},{"type":"string","name":"a.c"// Store data in the a.c path.}] **Note** For the column parameter, you must specify the type and name fields.	Yes	No default value
dataPath	The path that is used to store the JSON-formatted data.	No	No default value
method	The request method. Valid values: post and put.	Yes	No default value
customHeader	The header information transferred to the RESTful API.	No	No default value
authType	The authentication method. Valid values: - Basic Auth: basic authentication If the data source API supports username and password-based authentication, you can select Basic Auth and configure the username and password to be used for authentication. During data integration, the username and password are transferred to the RESTful API URL for authentication. The data source is connected only after the authentication succeeds. - Token Auth: token-based authentication If the data source API supports token-based authentication, you can select Token Auth and configure a fixed token value to be used for authentication. During data integration, the token is contained in the request header, such as {"Authorization":"Bearer TokenXXXXXX"}, and transferred to the RESTful API URL for authentication. The data source is connected only after the authentication succeeds. - Aliyun API Signature: Alibaba Cloud API signature-based authentication If the following conditions are met, you can select Aliyun API Signature and configure the AccessKey ID and AccessKey secret to be used for authentication: The data source you want to connect is an Alibaba Cloud service and the API of this service supports AccessKey pair-based authentication.	No	No default value
authUsername/authPassword	The username and password used for basic authentication.	No	No default value
authToken	The token used for token-based authentication.	No	No default value
accessKey/accessSecret	The AccessKey pair used for Alibaba Cloud API signature-based authentication.	No	No default value

Parameter	Description	Required	Default value
batchSize	The maximum number of data records that can be transferred in each request when the dataMode parameter is set to multiData.	Yes	512

Configure RestAPI Writer by using the codeless UI

1. Configure data sources.

   Configure Source and **Target** for the synchronization node.

   * Connection: RestAPI | doc_test_api_02

   * Request Method: ● POST ○ PUT

   * Data structure of request parameters: ○ Single Data ● Array Data

   BatchSize: 1024

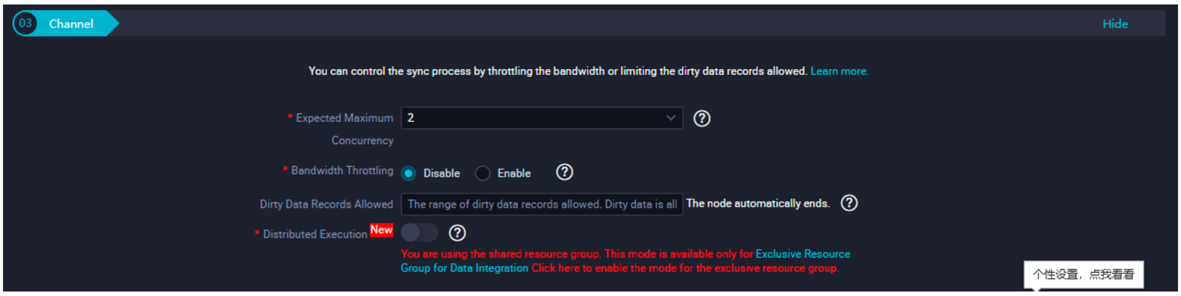
   json path to store data:

   Header: {"xxx-header":"xxx"}

Parameter	Description
Connection	Select **RestAPI** from the left-side drop-down list and a data source from the right-side drop-down list in the Target section.
Request Method	This parameter is equivalent to the method parameter that is described in the preceding section.
Data structure of request parameters	This parameter is equivalent to the dataMode parameter that is described in the preceding section.
BatchSize	This parameter is equivalent to the batchSize parameter that is described in the preceding section.
json path to store data	This parameter is equivalent to the dataPath parameter that is described in the preceding section.
Header	This parameter is equivalent to the customHeader parameter that is described in the preceding section.

2.

3. Configure channel control policies.

   You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

   * Expected Maximum Concurrency: 2

   * Bandwidth Throttling: ● Disable ○ Enable

   Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

   * Distributed Execution New

   You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.

Parameter	Description
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	This parameter is not supported for synchronization nodes that use RestAPI Writer.

Configure RestAPI Reader by using the code editor

In the following code, a synchronization node is configured to write data to a RestAPI data source:

```
{
    "type":"job",
    "version":"2.0",
    "steps":[
        {
            "stepType":"stream",
            "parameter":{
            },
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"restapi",
            "parameter":{
                "url":"http://127.0.0.1:5000/writer1",
                "dataMode":"oneData",
                "responseType":"json",
                "column":[
                    {
                        "type":"long", // Store data in the a.b path.
                        "name":"a.b"
                    },
                    {
                        "type":"string", // Store data in the a.c path.
                        "name":"a.c"
                    }
                ],
                "method":"post",
                "defaultHeader":{
                    "X-Custom-Header":"test header"
                },
                "customHeader":{
                    "X-Custom-Header2":"test header2"
                },
                "parameters":"abc=1&amp;def=1",
                "batchSize":256
            },
            "name":"restapiwriter",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0" // The maximum number of dirty data records allowed.
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.39. SAP HANA Writer

This topic describes the parameters that are supported by SAP HANA Writer and how to configure SAP HANA Writer by using the codeless user interface (UI) and code editor.

Context

SAP HANA Writer writes data to tables stored in SAP HANA databases. SAP HANA Writer connects to a remote SAP HANA database by using Java Database Connectivity (JDBC) and executes the `INSERT INTO` or `REPLACE INTO` statement to write data to the SAP HANA database. SAP HANA uses the InnoDB engine so that data is written to the database in batches.

SAP HANA Writer can also be used as a data migration tool by users such as database administrators. SAP HANA Writer obtains data from a reader and writes the data to the destination database based on the value of the writeMode parameter.

> ⓘ **Note** A synchronization node that uses SAP HANA Writer must have at least the permissions to execute the `INSERT INTO or REPLACE INTO` statement. Whether other permissions are required depends on the SQL statements that you specify in the preSql and postSql parameters when you configure the node.

Parameters

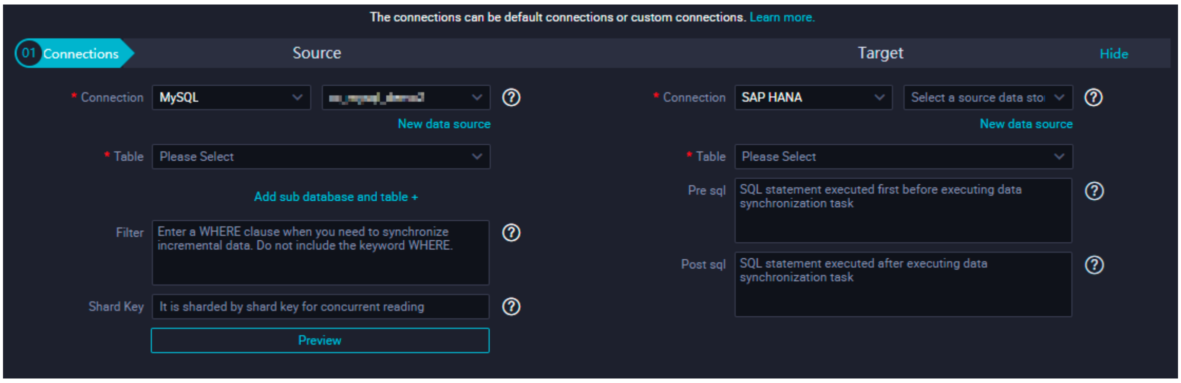
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table to which you want to write data.	Yes	No default value
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as `"column": ["id", "name", "age"]`. If you want to write data to all the columns in the destination table, set this parameter to an asterisk (*), such as `"column": ["*"]`.	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor. For example, you can set this parameter to the following SQL statement that is used to delete outdated data: `truncate table tablename` ⓘ **Note** If you specify multiple SQL statements, whether all the statements can be successfully executed cannot be ensured.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor. For example, you can set this parameter to the `alter table tablenameadd colname timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP` SQL statement that is used to add a timestamp. ⓘ **Note**	No	No default value

Parameter	Description	Required	Default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and SAP HANA and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1024*

Configure SAP HANA Writer by using the codeless UI

1. Configure data sources.

   Configure Source and **Target** for the synchronization node.



The connections can be default connections or custom connections. Learn more.

01 Connections

Source

* Connection: MySQL [redacted]

New data source

* Table: Please Select

Add sub database and table +

Filter: Enter a WHERE clause when you need to synchronize incremental data. Do not include the keyword WHERE.

Shard Key: It is sharded by shard key for concurrent reading

Preview

Target

Hide

* Connection: SAP HANA, Select a source data sto

New data source

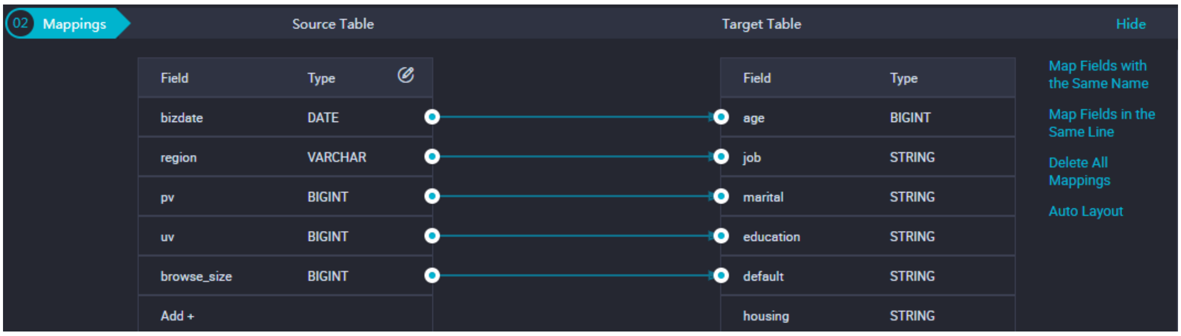
* Table: Please Select

Pre sql: SQL statement executed first before executing data synchronization task

Post sql: SQL statement executed after executing data synchronization task

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table to which you want to write data. This parameter is equivalent to the table parameter that is described in the preceding section.
Pre sql	The SQL statement that you want to execute before the synchronization node is run. This parameter is equivalent to the preSql parameter that is described in the preceding section.
Post sql	The SQL statement that you want to execute after the synchronization node is run. This parameter is equivalent to the postSql parameter that is described in the preceding section.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.

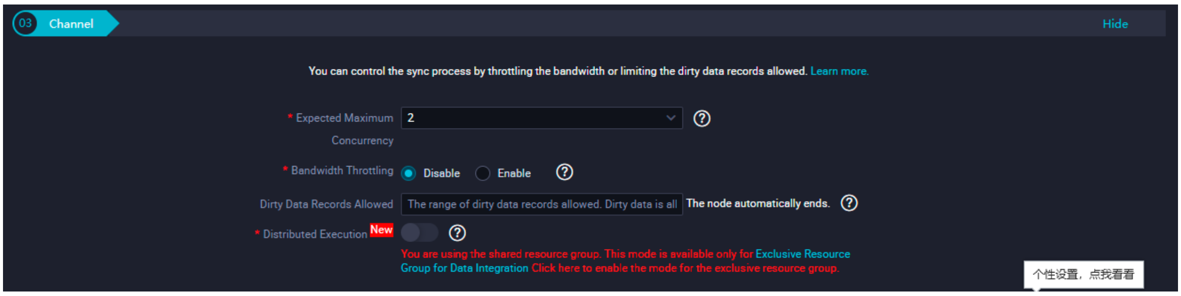


Field	Type	Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
Add +		housing	STRING

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.

Operation	Description
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2 ⓘ

* Bandwidth Throttling: ● Disable ○ Enable ⓘ

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends. ⓘ

* Distributed Execution New ⓘ

You are using the shared resource group. This mode is available only for Exclusive Resource Group for Data Integration Click here to enable the mode for the exclusive resource group.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	This parameter is not supported for synchronization nodes that use RestAPI Writer.

Configure SAP HANA Writer by using the code editor

In the following code, a synchronization node is configured to write data to an SAP HANA database:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"saphana",// The writer type.
            "parameter":{
                "postSql":[],// The SQL statement that you want to execute after the synchronization node is ru
n.
                "datasource":"",// The name of the data source.
                "column":[// The names of the columns to which you want to write data.
                    "id",
                    "value"
                ],
                "batchSize":1024,// The number of data records to write at a time.
                "table":"",// The name of the table to which you want to write data.
                "preSql":[
                     "delete from XXX;" // The SQL statement that you want to execute before the synchronizatio
n node is run.
                   ]
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{// The maximum number of dirty data records allowed.
            "record":"0"
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that
bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps p
arameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.40. KingbaseES Writer

This topic describes the parameters that are supported by KingbaseES Writer and how to configure KingbaseES Writer by using the codeless user interface (UI) and code editor.

Context

KingbaseES Writer writes data to tables stored in KingbaseES databases. KingbaseES Writer connects to a remote KingbaseES database by using Java Database Connectivity (JDBC) and executes the `INSERT INTO` or `REPLACE INTO` statement to write data to the KingbaseES database. KingbaseES uses the InnoDB engine so that data is written to the database in batches.

KingbaseES Writer can also be used as a data migration tool by users such as database administrators. KingbaseES Writer obtains data from a reader and writes the data to the destination database based on the value of the writeMode parameter.

> ⓘ **Note** A synchronization node that uses KingbaseES Writer must have at least the permissions to execute the `INSERT INTO or REPLACE INTO` statement. Whether other permissions are required depends on the SQL statements that you specify in the preSql and postSql parameters when you configure the node.

Parameters

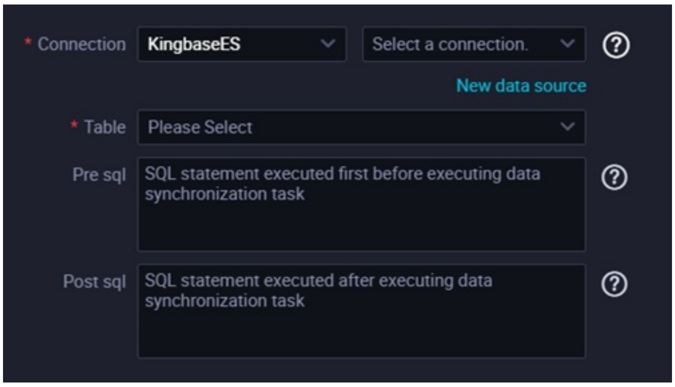
Parameter	Description	Required	Default value
datasource	The name of the data source. It must be the same as the name of the added data source. You can add data sources by using the code editor.	Yes	No default value
table	The name of the table to which you want to write data.	Yes	No default value
column	The names of the columns to which you want to write data. Separate the names with commas (,), such as `"column": ["id", "name", "age"]` . If you want to write data to all the columns in the destination table, set this parameter to an asterisk (*), such as `"column": ["*"]` .	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor. For example, you can set this parameter to the following SQL statement that is used to delete outdated data: `truncate table tablename` ⓘ **Note** If you specify multiple SQL statements, whether all the statements can be successfully executed cannot be ensured.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run. You can execute only one SQL statement on the codeless UI and multiple SQL statements in the code editor. For example, you can set this parameter to the `alter table tablenameadd colname timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP` SQL statement that is used to add a timestamp. ⓘ **Note**	No	No default value

Parameter	Description	Required	Default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and SAP HANA and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1024*

Configure KingbaseES Writer by using the codeless UI

1. Configure data sources.

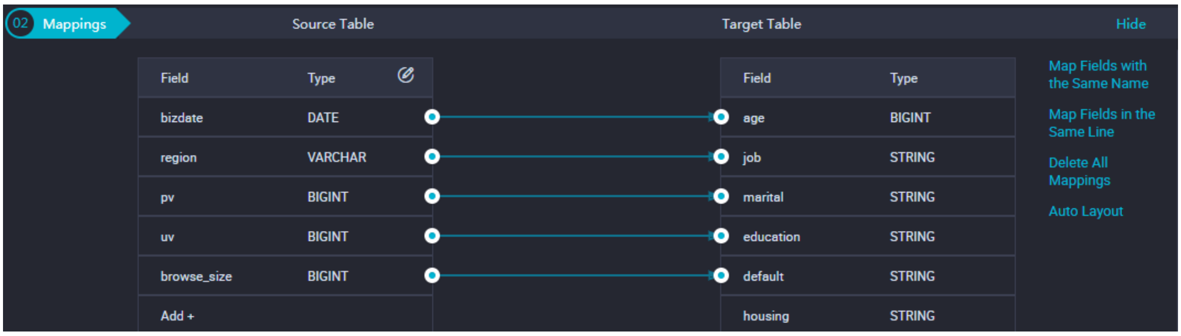
   Configure Source and **Target** for the synchronization node.



* Connection	KingbaseES Select a connection. New data source
* Table	Please Select
Pre sql	SQL statement executed first before executing data synchronization task
Post sql	SQL statement executed after executing data synchronization task

Parameter	Description
Connection	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Table	The name of the table to which you want to write data. This parameter is equivalent to the table parameter that is described in the preceding section.
Pre sql	The SQL statement that you want to execute before the synchronization node is run. This parameter is equivalent to the preSql parameter that is described in the preceding section.
Post sql	The SQL statement that you want to execute after the synchronization node is run. This parameter is equivalent to the postSql parameter that is described in the preceding section.

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.

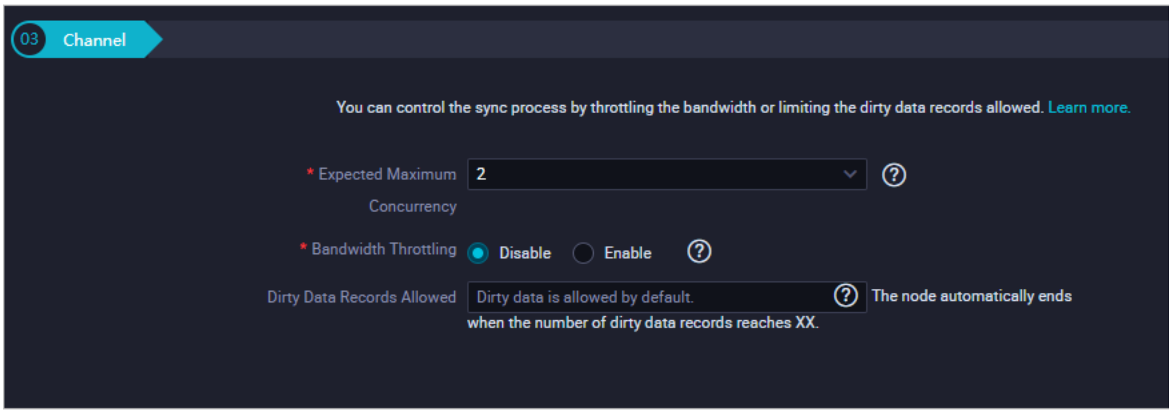


Field	Type	Field	Type
bizdate	DATE	age	BIGINT
region	VARCHAR	job	STRING
pv	BIGINT	marital	STRING
uv	BIGINT	education	STRING
browse_size	BIGINT	default	STRING
Add +		housing	STRING

Operation	Description

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: Dirty data is allowed by default. The node automatically ends when the number of dirty data records reaches XX.

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node uses to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.

Configure KingbaseES Writer by using the code editor

In the following code, a synchronization node is configured to write data to a KingbaseES database:

```
{
    "type":"job",
    "version":"2.0",// The version number.
    "steps":[
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"kingbasees",// The writer type.
            "parameter":{
                "postSql":[],// The SQL statement that you want to execute after the synchronization node is run.
                "datasource":"",// The name of the data source.
                "column":[// The names of the columns to which you want to write data.
                    "id",
                    "value"
                ],
                "batchSize":1024,// The number of data records to write at a time.
                "table":"",// The name of the table to which you want to write data.
                "preSql":[
                     "delete from XXX;" // The SQL statement that you want to execute before the synchronization node is run.
                   ]
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{// The maximum number of dirty data records allowed.
            "record":"0"
        },
        "speed":{
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":1, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

5.3.41. DB Writer

This topic describes the data types and parameters that are supported by DB Writer and how to configure DB Writer by using the code editor.

Notice DB Writer supports only exclusive resource groups for Data Integration, but not the shared resource group or custom resource groups for Data Integration. For more information, see Create and use an exclusive resource group for Data Integration and Create a custom resource group for Data Integration.

Background information

DB Writer writes data to tables stored in primary databases. DB Writer obtains data from a reader, connects to a remote database by using Java Database Connectivity (JDBC), and then executes an `INSERT INTO` statement to write data to the database. DB Writer is commonly used to write data to relational databases. To enable DB Writer to write data to a relational database, you must register the driver for the relational database.

DB Writer is designed for extract, transform, load (ETL) developers to import data in data warehouses to relational databases. DB Writer can also be used as a data migration tool by users such as database administrators.

DB Writer supports most data types of common relational databases, such as numeric and string data types. Make sure that the data types of your database are supported.

Parameters

Parameter	Description	Required	Default value
jdbcUrl	The JDBC URL of the destination database. The format of the JDBC URL that you specify must follow the official specifications of the database. You can also specify the information of the attachment facility. The format varies based on the database type. Data Integration selects the most suitable driver based on the format and uses this driver to write data to the destination database. - Format for DM databases: `jdbc:dm://IP address:Port number/database` - Format for Db2 databases: `jdbc:db2://IP address:Port number/database` - Format for PPAS databases: `jdbc:edb://IP address:Port number/database`	Yes	No default value
username	The username that is used to connect to the destination database.	Yes	No default value
password	The password that is used to connect to the destination database.	Yes	No default value
table	The name of the table to which you want to write data. If the table uses the default schema for the destination database, the value of this parameter consists of only the name of the table. If the table uses a custom schema, the value of this parameter consists of two parts: the name of the custom schema and the name of the table. Specify the two parts in the `Schema name.Table name` format.	Yes	No default value
column	The names of the columns to which you want to write data. Separate the columns with commas (,). **Note** We recommend that you do not leave this parameter empty.	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to delete outdated data. You can execute only one SQL statement in a transaction. **Note** If you specify multiple SQL statements, the statements are not executed in the same transaction.	No	No default value

Parameter	Description	Required	Default value
postSql	The SQL statement that you want to execute after the synchronization node is run. For example, you can set this parameter to the SQL statement that is used to add a timestamp. You can execute only one SQL statement in a transaction. **Note** If you specify multiple SQL statements, the statements are not executed in the same transaction.	No	No default value
batchSize	The number of data records to write at a time. Set this parameter to an appropriate value based on your business requirements. This greatly reduces the interactions between Data Integration and the destination database and increases throughput. If you set this parameter to an excessively large value, an out of memory (OOM) error may occur during data synchronization.	No	*1024*

Configure DB Writer by using the code editor

In the following code, a synchronization node is configured to write data to a database. For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

```
{
    "type": "job",
    "steps": [
        {
            "stepType": "oracle",
            "parameter": {
                "datasource": "aaa",
                "column": [
                    "PROD_ID",
                  "name"
                ],
                "where": "",
                "splitPk": "",
                "encoding": "UTF-8",
                "table": "PENGXI.SALES"
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "rdbms",
            "parameter": {
                "connection": [
                    {
                        "jdbcUrl": "jdbc:dm://ip:port/database",
                        "table": [
                            "table"
                        ]
                    }
                ],
                "username": "username",
                "password": "password",
                "table": "table",
                "column": [
                   "id",
                  "name"
                ],
                "preSql": [
                    "delete from XXX;"
                ]
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": ""
        },
        "speed": {
            "throttle":true,// Specifies whether to enable bandwidth throttling. The value false indicates that bandwidth throttling is disabled, and the value true indicates that bandwidth throttling is enabled. The mbps parameter takes effect only when the throttle parameter is set to true.
            "concurrent":2, // The maximum number of parallel threads.
            "mbps":"12"// The maximum transmission rate.
        }
    }
}
```

To enable DB Writer to support a new type of database, perform the following steps:

1. Go to the directory of DB Writer, which is *${DATAX_HOME}/plugin/writer/DB Writer*. In the directory, *${DATAX_HOME}* indicates the main directory of Data Integration.
2. Open the *plugin.json* file in the directory of DB Writer and add the driver of the database to the drivers array in the file. During the running of a synchronization node, DB Writer automatically selects the most suitable driver to connect to the database.

```
{
    "name": "DB Writer",
    "class": "com.alibaba.datax.plugin.reader.DB Writer.DB Writer",
    "description": "useScene: prod. mechanism: Jdbc connection using the database, execute select sql, retrieve data from the ResultSet. warn: The more you know about the database, the less problems you encounter.",
    "developer": "alibaba",
    "drivers": [
        "dm.jdbc.driver.DmDriver",
        "com.ibm.db2.jcc.DB2Driver",
        "com.sybase.jdbc3.jdbc.SybDriver",
        "com.edb.Driver"
    ]
}
```

3. Add the package of the driver to the *libs* directory in the directory of DB Writer.

```
$tree
.
|-- libs
|   |-- Dm7JdbcDriver16.jar
|   |-- commons-collections-3.0.jar
|   |-- commons-io-2.4.jar
|   |-- commons-lang3-3.3.2.jar
|   |-- commons-math3-3.1.1.jar
|   |-- datax-common-0.0.1-SNAPSHOT.jar
|   |-- datax-service-face-1.0.23-20160120.024328-1.jar
|   |-- db2jcc4.jar
|   |-- druid-1.0.15.jar
|   |-- edb-jdbc16.jar
|   |-- fastjson-1.1.46.sec01.jar
|   |-- guava-r05.jar
|   |-- hamcrest-core-1.3.jar
|   |-- jconn3-1.0.0-SNAPSHOT.jar
|   |-- logback-classic-1.0.13.jar
|   |-- logback-core-1.0.13.jar
|   |-- plugin-rdbms-util-0.0.1-SNAPSHOT.jar
|   `-- slf4j-api-1.7.10.jar
|-- plugin.json
|-- plugin_job_template.json
`-- DB Writer-0.0.1-SNAPSHOT.jar
```

5.3.42. StarRocks Writer

This topic describes the data types and parameters that are supported by StarRocks Writer and how to configure StarRocks Writer by using the codeless user interface (UI) and code editor.

Limits

E-MapReduce (EMR) StarRocks V2.1 is supported. For more information about EMR StarRocks.

How it works

You can use StarRocks Writer to write data to a destination table in a StarRocks database. Data is written to a destination table as CSV files in batches by using the Stream Load import method.

Data types

StarRocks Writer supports most StarRocks data types, including numeric, STRING, and DATE data types.

Parameters

Parameter	Description	Required	Default value
datasource	The name of the StarRocks data source.	Yes	No default value
selectedDatabase	The name of the StarRocks database.	No	The name of the database that is configured in the StarRocks data source

Parameter	Description	Required	Default value
loadProps	The request parameters for the StarRocks Stream Load import method. If you want to import data as CSV files by using the Stream Load import method, you can configure request parameters. If you have no special requirements, set the parameter to {}. Request parameters that you can configure for the Stream Load import method: • column_separator: specifies the column delimiter of a CSV file. The default value is \t. • row_delimiter: specifies the row delimiter of a CSV file. The default value is \n. • If the data you want to write to StarRocks contains \t or \n, you must use other characters as delimiters. Example: `null`	Yes	No default value
column	The names of the columns to which you want to write data.	Yes	No default value
loadUrl	The URL of a StarRocks frontend node. The URL consists of the IP address of the frontend node and the HTTP port number. The default HTTP port number is 8030. If you specify URLs for multiple frontend nodes, separate them with commas (,).	Yes	No default value
table	The name of the table to which you want to write data.	Yes	No default value
preSql	The SQL statement that you want to execute before the synchronization node is run. For example, you can set this parameter to the TRUNCATE TABLE tablename statement to delete outdated data.	No	No default value
postSql	The SQL statement that you want to execute after the synchronization node is run.	No	No default value

Configure StarRocks Writer by using the codeless UI

Create a synchronization node and configure the node. For more information, see Configure a synchronization node by using the codeless UI.

You must perform the following steps on the configuration tab of the synchronization node:

1. Configure data sources.

   Configure **Source** and **Target** for the synchronization node.

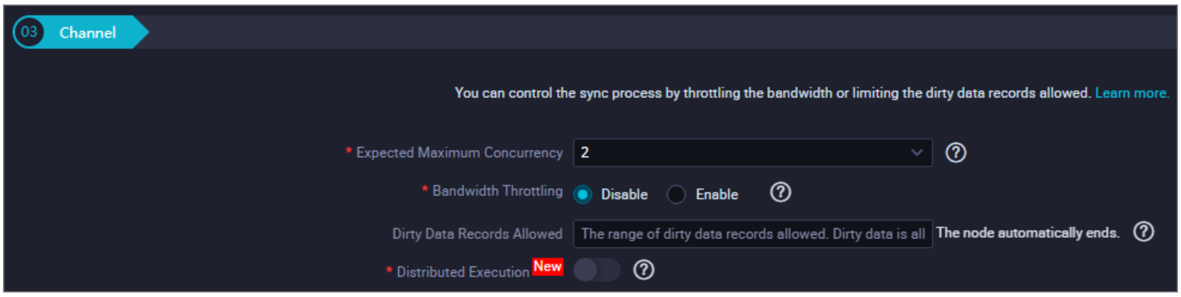
Parameter	Description
Data source	The name of the data source to which you want to write data. This parameter is equivalent to the datasource parameter that is described in the preceding section.
Database	The name of the database to which you want to write data. The database is the one that is configured in the Java Database Connectivity (JDBC) API that is used to access the StarRocks data source. This parameter is equivalent to the selectedDatabase parameter that is described in the preceding section.
Table	The name of the table to which you want to write data. This parameter is equivalent to the table parameter that is described in the preceding section.
Pre sql	The SQL statement that you want to execute before the synchronization node is run. This parameter is equivalent to the preSql parameter that is described in the preceding section.

Parameter	Description
Post sql	The SQL statement that you want to execute after the synchronization node is run. This parameter is equivalent to the postSql parameter that is described in the preceding section.
LoadUrls	The URL of a StarRocks frontend node. The URL consists of the IP address of the frontend node and the HTTP port number. The default HTTP port number is 8030. If you specify URLs for multiple frontend nodes, separate them with commas (,). This parameter is equivalent to the loadUrl parameter that is described in the preceding section.
StreamLoad Request Parameters	The parameters for data import if you want to import data as CSV files by using the Stream Load import method. This parameter is equivalent to the loadProps parameter that is described in the preceding section. If you have no special requirements, set the parameter to {}. Request parameters that you can configure for the Stream Load import method: ○ column_separator: specifies the column delimiter of a CSV file. The default value is \t. ○ row_delimiter: specifies the row delimiter of a CSV file. The default value is \n. ○ If the data you want to write to StarRocks contains \t or \n, you must use other characters as delimiters. Example: `null`

2. Configure field mappings. This operation is equivalent to setting the column parameter that is described in the preceding section. Fields in the source on the left have a one-to-one mapping with fields in the destination on the right.

Operation	Description
Map Fields with the Same Name	Click **Map Fields with the Same Name** to establish mappings between fields with the same name. The data types of the fields must match.
Map Fields in the Same Line	Click **Map Fields in the Same Line** to establish mappings between fields in the same row. The data types of the fields must match.
Delete All Mappings	Click **Delete All Mappings** to remove the mappings that are established.
Auto Layout	Click Auto Layout. Then, the system automatically sorts the fields based on specific rules.
Change Fields	Click the Change Fields icon. In the Change Fields dialog box, you can manually edit the fields in the source table. Each field occupies a row. The first and the last blank rows are included, whereas other blank rows are ignored.
Add	Click **Add** to add a field. You can add fields of the following types: ○ You can enter constants. Each constant must be enclosed in single quotation marks ('), such as 'abc' and '123'. ○ You can use scheduling parameters, such as ${bizdate}. ○ You can enter functions that are supported by relational databases, such as now() and count(1). ○ If the field that you entered cannot be parsed, the value of Type for the field is Unidentified.

3. Configure channel control policies.



You can control the sync process by throttling the bandwidth or limiting the dirty data records allowed. Learn more.

* Expected Maximum Concurrency: 2

* Bandwidth Throttling: ● Disable ○ Enable

Dirty Data Records Allowed: The range of dirty data records allowed. Dirty data is all The node automatically ends.

* Distributed Execution New

Parameter	Description
Expected Maximum Concurrency	The maximum number of parallel threads that the synchronization node can use to read data from the source or write data to the destination. You can configure the parallelism for the synchronization node on the codeless UI.
Bandwidth Throttling	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and specify a maximum transmission rate to prevent heavy read workloads on the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to an appropriate value based on the configurations of the source.
Dirty Data Records Allowed	The maximum number of dirty data records allowed.
Distributed Execution	The distributed execution mode that allows you to split your node into pieces and distribute them to multiple Elastic Compute Service (ECS) instances for parallel execution. This speeds up synchronization. If you use a large number of parallel threads to run your synchronization node in distributed execution mode, excessive access requests are sent to the data sources. Therefore, before you use the distributed execution mode, you must evaluate the access load on the data sources. You can enable this mode only if you use an exclusive resource group for Data Integration. For more information about exclusive resource groups for Data Integration, see Overview and Create and use an exclusive resource group for Data Integration.

Configure StarRocks Writer by using the code editor

For more information about how to configure a synchronization node by using the code editor, see Create a synchronization node by using the code editor.

In the following code, a synchronization node is configured to write data to a StarRocks database. For more information about parameters, see the preceding parameter description.

```
{
    "stepType": "starrocks",
    "parameter": {
        "selectedDatabase": "didb1",
        "loadProps": {
            "row_delimiter": "\\x02",
            "column_separator": "\\x01"
        },
        "datasource": "starrocks_public",
        "column": [
            "id",
            "name"
        ],
        "loadUrl": [
            "1.1.1.1:8030"
        ],
        "table": "table1",
        "preSql": [
            "truncate table table1"
        ],
        "postSql": [
        ]
    },
    "name": "Writer",
    "category": "writer"
}
```