Alibaba Cloud

Data Transmission Service

Data Synchronization

Document Version: 20200820

(-) 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.

- 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 Cloudauthorized 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 |
|----------------|---|---|
| <u> Danger</u> | 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 listinstanceid 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.Data synchronization procedure list | 07 |
|--|-----|
| 2.Replication topologies | 10 |
| 3.ETL features | 13 |
| 3.1. Data type conversion | 13 |
| 3.2. Object name mapping | 16 |
| 3.3. Data filtering | 19 |
| 4.Synchronization task management | 22 |
| 4.1. Check the synchronization performance | 22 |
| 4.2. Add an object to a data synchronization task | 23 |
| 4.3. Remove an object from a data synchronization task | 25 |
| 4.4. Stop a data synchronization task | 27 |
| 4.5. Reset a data synchronization task | 27 |
| 5.Precheck troubleshooting | 29 |
| 5.1. Troubleshoot precheck failures | 29 |
| 6.Synchronize data between MySQL databases | 33 |
| 6.1. Configure two-way data synchronization between Apsara | 33 |
| 6.2. Configure one-way data synchronization between Apsar | 43 |
| 6.3. Synchronize data from a user-created MySQL database | 50 |
| 6.4. Synchronize data from a user-created MySQL database | 57 |
| 6.5. Synchronize data from an ApsaraDB RDS for MySQL inst | 64 |
| 6.6. Synchronize data between ApsaraDB RDS for MySQL inst | 72 |
| 7.Synchronize data from MySQL databases to other databases | 80 |
| 7.1. Synchronize data from an ApsaraDB RDS for MySQL insta | 80 |
| 7.2. Synchronize data from an ApsaraDB RDS for MySQL inst | 87 |
| 7.3. Synchronize data from an ApsaraDB RDS for MySQL inst | 94 |
| 7.4. Synchronize data from a user-created MySQL database | 100 |

| 7.5. Synchronize data from a user-created MySQL database | 107 |
|---|-------|
| 7.6. Synchronize data from a user-created MySQL database | - 114 |
| 7.7. Synchronize data from an ApsaraDB RDS MySQL instance | - 122 |
| 7.8. Use alternative naming for system-created columns | - 129 |
| 7.9. Synchronize data from a user-created MySQL database t | - 131 |
| 7.10. Synchronize data from an ApsaraDB RDS for MySQL ins | - 138 |
| 7.11. Synchronize data from a user-created MySQL database | - 144 |
| 7.12. Synchronize data from a user-created MySQL database | - 149 |
| 8.Synchronize data to or from an Apsara PolarDB cluster | - 156 |
| 8.1. Configure two-way data synchronization between Apsara | - 156 |
| 8.2. Configure one-way data synchronization between Apsar | - 166 |
| 8.3. Synchronize data between PolarDB clusters compatible | - 173 |
| 8.4. Synchronize data from a PolarDB cluster compatible wit | - 180 |
| 8.5. Synchronize data from an Apsara PolarDB for MySQL cl | - 185 |
| 8.6. Synchronize data from an Apsara PolarDB for MySQL cl | - 192 |
| 8.7. Synchronize data from an Apsara PolarDB for MySQL cl | - 199 |
| 8.8. Synchronize data from an Apsara PolarDB for MySQL cl | - 205 |
| 8.9. Synchronize data from an Apsara PolarDB for MySQL cl | - 212 |
| 8.10. Synchronize data from a user-created MySQL database | - 219 |
| 8.11. Synchronize data from a user-created MySQL database | - 227 |
| 9.Synchronize data to or from a DRDS instance | - 235 |
| 9.1. Synchronize data between DRDS instances | - 235 |
| 9.2. Synchronize data from a DRDS instance to an AnalyticD | - 239 |
| 9.3. Synchronize data from a DRDS instance to an AnalyticD | - 245 |
| 10.Synchronize data between PostgreSQL databases | - 251 |
| 10.1. Synchronize data from an ApsaraDB RDS for PostgreSQ | - 251 |
| 10.2. Synchronize data from a user-created PostgreSQL data | - 257 |
| 11.Synchronize data between Redis databases | - 264 |

| 11.1. Configure one-way data synchronization between Apsa 2 | 264 |
|---|-----|
| 11.2. Configure two-way data synchronization between Apsa | 269 |
| 11.3. Synchronize data from a user-created Redis database 2 | 276 |
| 11.4. Synchronize data from a user-created Redis database | 282 |
| 11.5. Synchronize data from a user-created Redis cluster to | 289 |
| 11.6. Synchronize data from a Codis cluster hosted on ECS t | 295 |
| 11.7. Synchronize data from a Twemproxy Redis cluster host | 303 |

1.Data synchronization procedure list

The data synchronization mode of Data Transmission Service (DTS) replicates ongoing changes between two data stores. This mode is typically used for OLTP-to-OLAP replications and disaster recovery-related replications. The following table lists the topics for configuring data synchronization for different combinations of sources and targets.

| Scenario | Торіс |
|--|---|
| | Configure two-way data synchronization between ApsaraDB RDS for MySQL instances |
| | Configure one-way data synchronization between ApsaraDB RDS for MySQL instances |
| | Synchronize data from a user-created MySQL database hosted on ECS to an ApsaraDB RDS for MySQL instance |
| Synchronize data between MySQL databases | Synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an ApsaraDB RDS for MySQL instance |
| | Synchronize data from an ApsaraDB RDS for MySQL instance to a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway |
| | Synchronize data between ApsaraDB RDS for MySQL instances that belong to different Alibaba Cloud accounts |
| | Synchronize data from an ApsaraDB RDS for MySQL instance to an Apsara PolarDB for MySQL cluster |
| | Synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for MySQL cluster |
| | Synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for PostgreSQL instance |
| | Synchronize data from a user-created MySQL database hosted on ECS to an AnalyticDB for PostgreSQL instance |
| | Synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an AnalyticDB for PostgreSQL instance |
| Synchronize data from a | Synchronize data from a user-created MySQL database hosted on ECS to Elasticsearch |
| MySQL database to a different type of database | Synchronize data from an ApsaraDB RDS MySQL instance to a MaxCompute project |
| | Synchronize data from a user-created MySQL database to a Message Queue for Apache Kafka instance |

| Scenario | Торіс |
|--|---|
| | Synchronize data from an ApsaraDB RDS for MySQL instance to a user-created Kafka cluster |
| | Synchronize data from a user-created MySQL database hosted on ECS to a user-created Kafka cluster |
| | Synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to a user-created Kafka cluster |
| | Configure two-way data synchronization between Apsara PolarDB for MySQL clusters |
| | Configure one-way data synchronization between Apsara PolarDB for MySQL clusters |
| | Synchronize data between PolarDB clusters compatible with Oracle |
| | Synchronize data from a PolarDB cluster compatible with Oracle to a user-created Kafka cluster |
| | Synchronize data from an Apsara PolarDB for MySQL cluster to an ApsaraDB RDS for MySQL instance |
| Synchronize data to or from an Apsara PolarDB cluster | Synchronize data from an Apsara PolarDB for MySQL cluster to an Elasticsearch instance |
| | Synchronize data from an Apsara PolarDB for MySQL cluster to a user-created Kafka cluster |
| | Synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for MySQL cluster |
| | Synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for PostgreSQL instance |
| | Synchronize data from a user-created MySQL database hosted on ECS to an Apsara PolarDB for MySQL cluster |
| | Synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an Apsara PolarDB for MySQL cluster |
| | Synchronize data between DRDS instances |
| | Synchronize data from a DRDS instance to an AnalyticDB for MySQL cluster |
| Synchronize data to or from a DRDS instance | Synchronize data from a DRDS instance to an AnalyticDB for PostgreSQL instance |

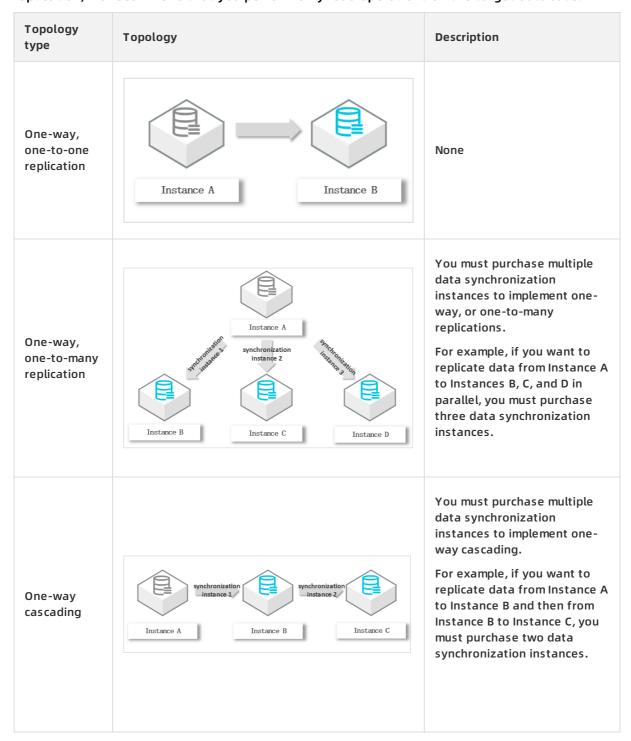
| Scenario | Торіс |
|---|---|
| | Configure one-way data synchronization between ApsaraDB for Redis instances |
| | Configure two-way data synchronization between ApsaraDB for Redis Enhanced Edition instances |
| | Synchronize data from a user-created Redis database hosted on ECS to an ApsaraDB for Redis instance |
| Synchronize data between Redis databases | Synchronize data from a user-created Redis database connected over Express Connect, VPN Gateway, or Smart Access Gateway to a user-created Redis database hosted on ECS |
| | Synchronize data from a user-created Redis cluster to an ApsaraDB for Redis cluster instance |
| | Synchronize data from a Codis cluster hosted on ECS to an ApsaraDB for Redis instance |
| | Synchronize data from a Twemproxy Redis cluster hosted on ECS to an ApsaraDB for Redis instance |
| Synchronize data between | Synchronize data from an ApsaraDB RDS for PostgreSQL instance to an AnalyticDB for PostgreSQL instance |
| PostgreSQL databases | Synchronize data from a user-created PostgreSQL database to an AnalyticDB for PostgreSQL instance |

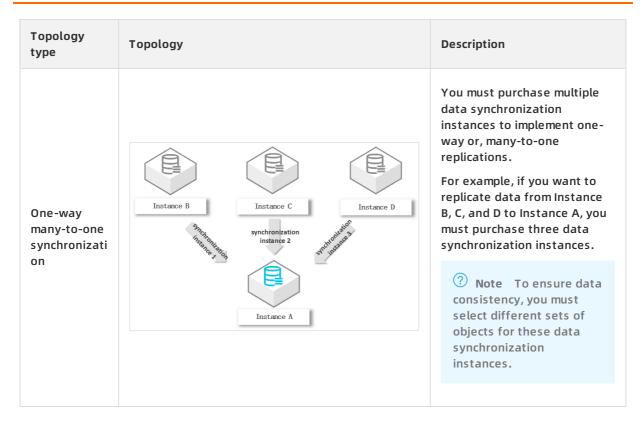
2. Replication topologies

The data synchronization mode of Data Transmission Service (DTS) supports multiple types of synchronization topologies. You can design and deploy your data synchronization instances based on your use cases.

One-way replication

To ensure data consistency between systems that are synchronized by using a one-way replication, we recommend that you perform only read operations on the target database.





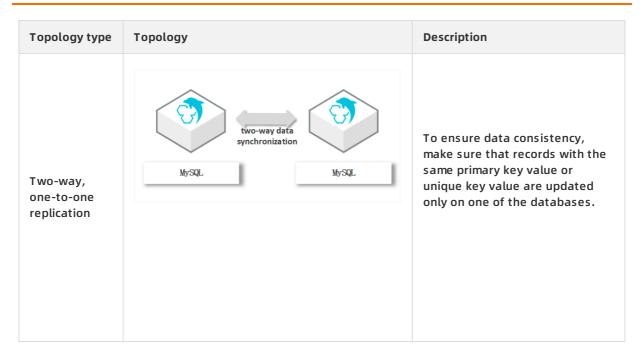
Two-way replication

DTS only supports two-way replication between two MySQL databases. DTS does not support two-way replication between multiple MySQL databases.



- For more information about how to configure two-way replication, see Configure two-way data synchronization between ApsaraDB RDS for MySQL instances.
- You can upgrade the synchronization topology of a data synchronization task from one-way to two-way. For more information, see Upgrade synchronization topology from one-way to two-way.

| Topology type | Topology | Description |
|---------------|----------|-------------|
|---------------|----------|-------------|



References

- Data synchronization mode
- Data synchronization procedure list

3.ETL features

3.1. Data type conversion

For a heterogeneous replication, Data Transmission Service converts columns to data types supported by the target database engine. The following tables list the mappings of source and target types for supported replication pairs.

MySQL-to-AnalyticDB

Source: a user-created MySQL database, ApsaraDB RDS for MySQL database, or Apsara PolarDB for MySQL

Target: an AnalyticDB for MySQL database

| MySQL data type | AnalyticDB for MySQL data type |
|--------------------|--------------------------------|
| BIGINT UNSIGNED | DECIMAL(20,0) |
| BINARY | VARBINARY |
| ВІТ | VARCHAR |
| BLOB | VARBINARY |
| CHAR | VARCHAR |
| DATE | DATE |
| DATETIME | DATETIME |
| DECIMAL | DECIMAL |
| DOUBLE | DOUBLE |
| DNUM | VARCHAR |
| FLOAT | FLOAT |
| GEOMETRY | VARBINARY |
| GEOMETRYCOLLECTION | VARBINARY |
| INT UNSIGNED | BIGINT |
| INTEGER | INT |
| JSON | JSON |
| LINESTRING | VARBINARY |
| LONGBLOB | VARBINARY |

| MySQL data type | AnalyticDB for MySQL data type |
|--------------------|--------------------------------|
| LONGTEXT | VARCHAR |
| MEDIUMBLOB | VARBINARY |
| MEDIUMINT | INT |
| MEDIUMINT UNSIGNED | INT |
| MEDIUMTEXT | VARCHAR |
| MULTILINESTRING | VARBINARY |
| MULTIPOINT | VARBINARY |
| MULTIPOLYGON | VARBINARY |
| NUMERIC | DECIMAL |
| POINT | VARBINARY |
| POLYGON | VARBINARY |
| SET | VARCHAR |
| SMALLINT UNSIGNED | INT |
| TEXT | VARCHAR |
| TIME | TIME |
| TIMESTAMP | TIMESTAMP |
| TINYBLOB | VARBINARY |
| TINYINT UNSIGNED | SMALLINT |
| TINYTEXT | VARCHAR |
| VARBINARY | VARBINARY |
| VARCHAR | VARCHAR |
| YEAR | INT |

MySQL-to-Elasticsearch

 $\textbf{Source:} \ a \ user-created \ MySQL \ database, ApsaraDB \ RDS \ for \ MySQL \ database, or \ Apsara \ PolarDB \ for \ MySQL \\$

Target: Elasticsearch

| MySQL data type | Elasticsearch data type | Description |
|--------------------|----------------------------|---|
| BIGINT | long | None |
| BINARY | binary | None |
| ВІТ | long | For BIT(1) columns in MySQL, we recommend that you use the BOOLEAN type in Elasticsearch. |
| BLOB | binary | None |
| BOOL or BOOLEAN | boolean | None |
| CHAR | text | None |
| DATE | date | The target DATE format is yyyy-MM-dd, in UTC time. For more information, see Date format mappings. |
| DATETIME | date | The target DATE format is yyyy-MM-dd'T'HH:mm:ss, in UTC time. A DATE field accurate to the microsecond is notated in the format of yyyy-MM-dd'T'HH:mm:ss.S. For more information, see Date format mappings. |
| DECIMAL or DEC | double | If the DECIMAL value contains a decimal point, we recommend that you use the TEXT type in Elasticsearch. |
| DOUBLE | double | None |
| FLOAT | float | None |
| GEOMETRY | geo_shape | None |
| GEOMETRYCOLLECTION | geo_shape | None |
| INT | integer | For UNSIGNED INT columns in MySQL, we recommend that you use the LONG type in Elasticsearch. |
| JSON | object | For a JSON object that carries only 1 bit of information, we recommend that you use the BOOLEAN type in Elasticsearch. |
| LINESTRING | geo_shape | None |
| LONGTEXT | text | None |
| MEDIUMINT | integer | None |
| MEDIUMTEXT | text | None |

| MySQL data type | Elasticsearch data type | Description |
|-----------------|----------------------------|--|
| MULTILINESTRING | geo_shape | None |
| MULTIPOINT | geo_shape | None |
| MULTIPOLYGON | geo_shape | None |
| POINT | geo_point | None |
| POLYGON | geo_shape | None |
| REAL | double | None |
| SMALLINT | short | For UNSIGNED SMALLINT columns in MySQL, we recommend that you use the INTEGER type in Elasticsearch. |
| TIME | date | The DATE format is HH:mm:ss, in UTC time. A DATE field accurateprecise to the microsecond is notated in the format of HH:mm:ss.S. For more information, see Date format mappings. |
| TIMESTAMP | date | The DATE format is yyyy-MM-dd'T'HH:mm:ss, in UTC time. A DATE field accurate to the microsecond is in the format of yyyy-MM-dd'T'HH:mm:ss.S. For more information, see Date format mappings. |
| TINYINT | short | For UNSIGNED TINYINT columns in MySQL, we recommend that you use the INTEGER type in Elasticsearch. |
| TINYTEXT | text | None |
| VARCHAR | text | None |
| YEAR | date | The target DATE format is yyyy, in UTC time. For more information, see Date format mappings. |

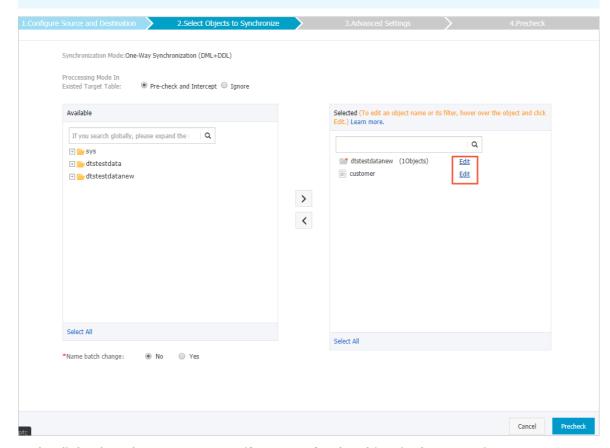
? Note To map a MySQL column to an Elasticsearch field using a data type different from the default data type, you must skip the initial schema synchronization and create the fields with the desired data types in Elasticsearch before you create a data synchronization task

3.2. Object name mapping

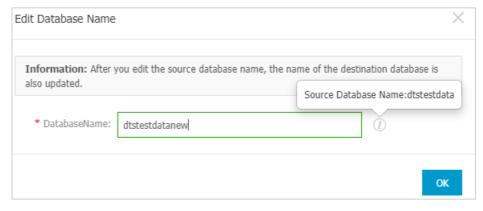
Data Transmission Service (DTS) allows you to configure name mappings between source and target columns, tables, or databases that have different names. You can define one-to-one mappings for individual objects for initial schema synchronization.

Procedure

- Note You can perform this operation only in the Select Objects to Synchronize step of creating a data synchronization task. If you change the name mapping after the data synchronization task has already started, the synchronization task may fail.
- 1. On the Select Objects to Synchronize page, move the required objects to the Selected list, move the pointer over a database or table, and then click Edit.
 - **?** Note Different database types support different objects. If Edit appears when you move the pointer over the target object, the operation is supported.

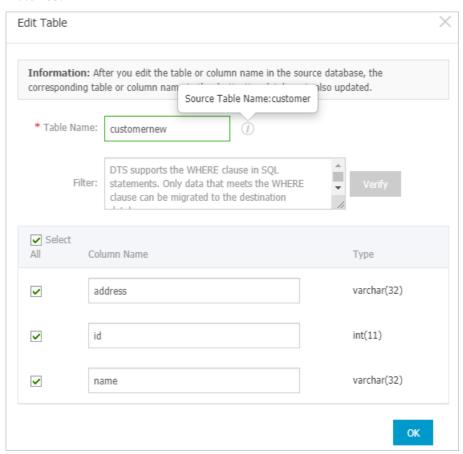


- 2. In the dialog box that appears, specify a name for the object in the target instance.
 - Database name mapping
 - In the **Edit Database Name** dialog box, enter the database name that you want to use in the target instance.



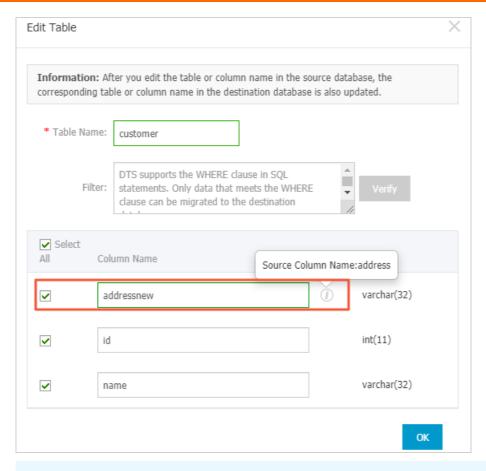
• Table name mapping

In the Edit Table dialog box, enter the table name that you want to use in the target instance.



Column name mapping

In the Edit Table dialog box, enter a new name for each column that you want to rename.



- ? Note In this step, you can deselect columns that do not need to be replicated.
- 3. Click OK.
- 4. Complete the other configurations and continue to create a task.

3.3. Data filtering

You can use SQL-like queries to filter the source table and replicate only records that match certain conditions. For example, you can use this feature to stream a table into several smaller tables.

Prerequisites

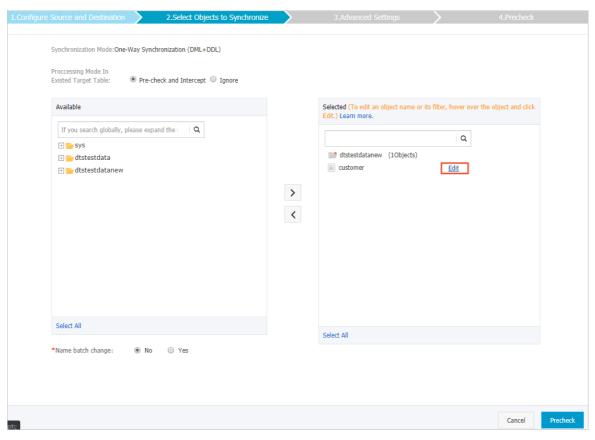
A data synchronization task is configured. The current step is **Select Objects to Synchronize**. For more information about how to create and configure a data synchronization task, see Create a data synchronization task.

Limits

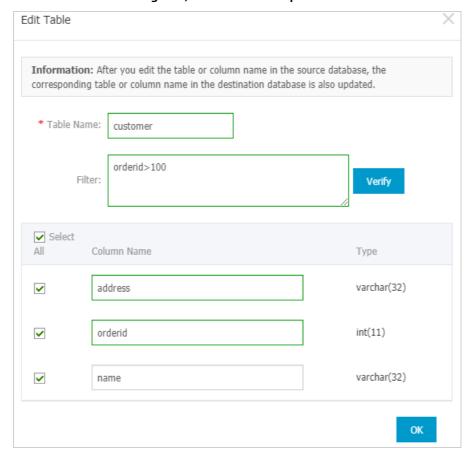
You can only perform filtering based on fields in the current table. You can not filter data by using fields in other tables.

Procedure

1. In the Select Objects to Synchronize step, move the required objects to the Selected list, move the pointer over a table, and then click Edit next to the object.



2. In the Edit Table dialog box, enter a filter expression in the Filter text area.



? Note

- The filter expression follows the syntax of the WHERE clause of most SQL languages. Do not include WHERE in the filter expression. The following operators are supported: = , ! = , < , and > . For example, orderid>100 .
- You can use apostrophes (') around object names for clearer delimitation. For example, address in ('sandiego', 'tokyo')
- 3. Click Verify to verify that the specified expression is valid.

? Note

- If the expression is valid, you receive an **Information** message that indicates the expression passed validation.
- If the expression is invalid, you receive an Error message that indicates there is an error in your expression. In this case, correct the expression and try the verification again.
- 4. Click OK.
- 5. Complete other configurations and continue to create a task.

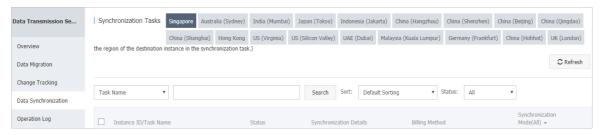
4. Synchronization task management

4.1. Check the synchronization performance

DTS provides three performance metrics: bandwidth, synchronization speed (TPS), and synchronization delay. You can monitor the progress of data synchronization tasks in the console by using the performance metrics.

Procedure

- 1. Log on to the DTS console.
- 2. In the left-side navigation pane, click Data Synchronization.
- 3. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



- 4. On the Synchronization Tasks page, click the ID of the destination instance.
- 5. In the left-side navigation pane, click Synchronization Performance.
- 6. Select a time range to view the trend charts of synchronization performance.



Performance metrics

| Performan ce metric | Description |
|---------------------------------------|---|
| Bandwidth | The bandwidth between the data writing module and the data pulling module. Unit: MB/s. |
| Synchroni zation speed (TPS) | The number of transactions that DTS synchronizes to the destination database per second. |
| Synchroni zation delay | The difference between the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. The unit is millisecond. |

4.2. Add an object to a data synchronization task

This topic describes how to add an object to a data synchronization task in the DTS console.

Prerequisites

- The data synchronization task is in the Synchronizing, Paused, or Synchronization Failed state.
- The source and destination databases are not in the process of upgrade, configuration change, network switchover, or cross-zone migration. This ensures that DTS can connect to the source and destination databases. This also ensures that DTS can read database and table information from the source database.

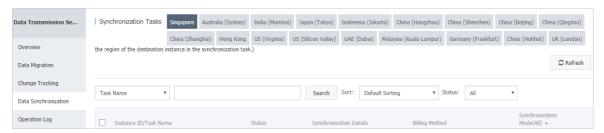
Precautions

The time when DTS synchronizes data of a new object depends on whether initial synchronization is configured for the data synchronization task.

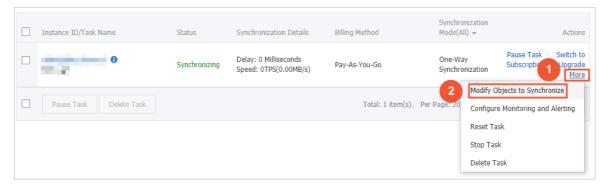
- If initial synchronization is not configured, DTS synchronizes data after incremental data is generated on the source instance.
- If initial synchronization is configured, DTS synchronizes schemas and historical data, and then synchronizes incremental data.

Procedure

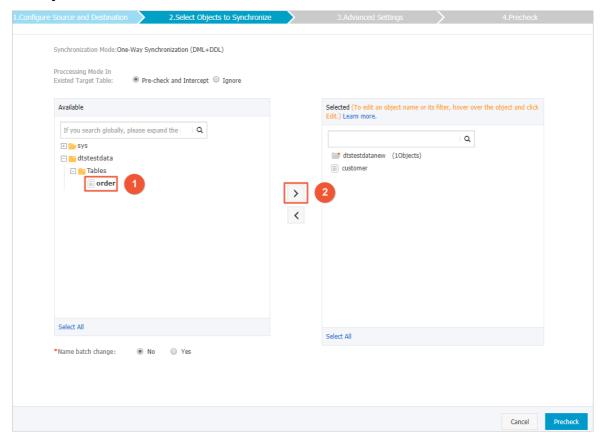
- 1. Log on to the DTS console.
- 2. In the left-side navigation pane, click Data Synchronization.
- 3. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



4. Find the data synchronization task and choose More > Modify Objects to Synchronize in the Actions column.



5. In the Available section, click the object that you want to add, and click the > icon to move the object to the Selected section.



6. Click Precheck.



- After you modify the objects to be synchronized, DTS performs a precheck on the data synchronization task. The data synchronization task can be started only after it passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 7. Start the data synchronization task.

4.3. Remove an object from a data synchronization task

This topic describes how to remove an object from a data synchronization task in the DTS console.

Prerequisites

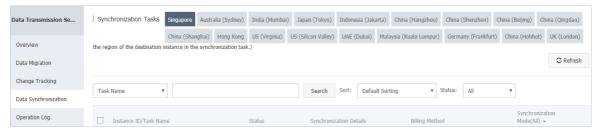
- The data synchronization task is in the Synchronizing, Paused, or Synchronization Failed state.
- The source and destination databases are not in the process of upgrade, configuration change, network switchover, or cross-zone migration. This ensures that DTS can connect to the source and destination databases. This also ensures that DTS can read database and table information from the source database.

Precautions

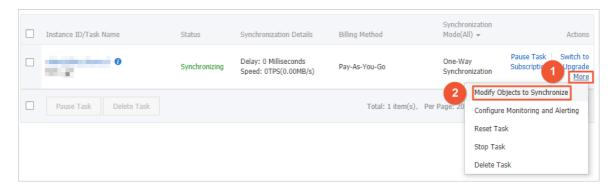
After an object is removed from a data synchronization task, the task no longer synchronizes incremental data of the object to the destination database.

Procedure

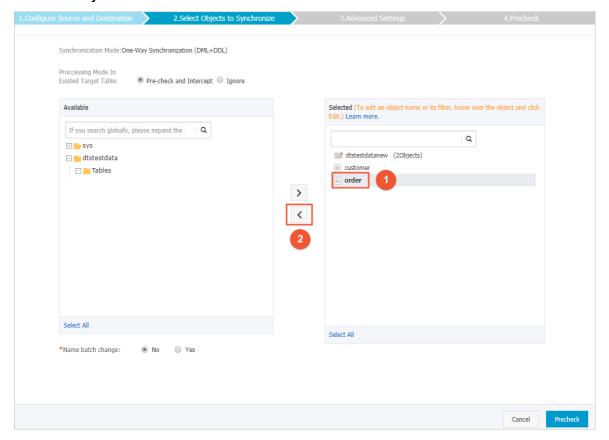
- 1. Log on to the DTS console.
- 2. In the left-side navigation pane, click Data Synchronization.
- 3. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



4. Find the data synchronization task and choose More > Modify Objects to Synchronize in the Actions column.



5. In the Selected section, click the object that you want to remove, and click the cicon to move the object to the Available section.



6. Click Precheck.

- ? Note
 - After you modify the objects to be synchronized, DTS performs a precheck on the data synchronization task. The data synchronization task can be started only after it passes the precheck.
 - If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 7. Start the data synchronization task.

4.4. Stop a data synchronization task

This topic describes how to stop a data synchronization task that you no longer need. After the data synchronization task is stopped, incremental data will not be synchronized to the destination database.

Prerequisites

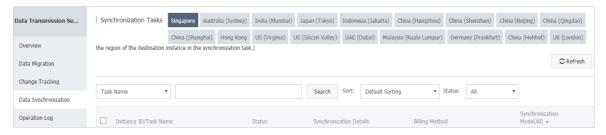
The data synchronization task is in the Synchronizing, Paused, Synchronization Failed, Performing Initial Synchronization, or Initial Synchronization Failed state.

Precautions

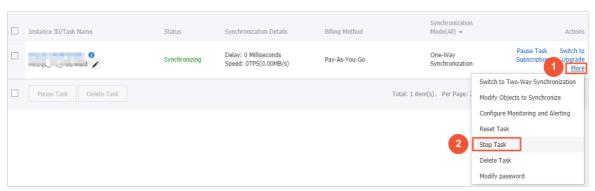
If a data synchronization task is stopped, the task enters the **Completed** state and no longer synchronizes incremental data from the source database to the destination database.

Procedure

- 1. Log on to the DTS console.
- 2. In the left-side navigation pane, click Data Synchronization.
- 3. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



4. Find the target data synchronization task, and choose More > Stop Task in the Actions column.



References

Release an instance

4.5. Reset a data synchronization task

This topic describes how to reset a data synchronization task. You can reset a data synchronization task to stop data synchronization or configure the task again.

Prerequisites

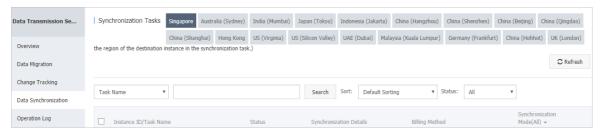
The data synchronization task is not in the Not Configured state.

Impacts on billing

- Subscription: no impact.
- Pay-as-you-go: The status of the data synchronization task changes to **Not Configured**. You are not billed for the task when it is in this state. The billing restarts only after you configure and start the data synchronization task.

Procedure

- 1. Log on to the DTS console.
- 2. In the left-side navigation pane, click Data Synchronization.
- 3. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



4. Find the target data synchronization task, and choose More > Reset Task in the Actions column.

• Warning Resetting a data synchronization task has the following impacts. Proceed with caution.

- The data synchronization task stops. Incremental data and schema changes in the source database are not synchronized to the destination database.
- The configurations of the data synchronization task are cleared. The status of the task changes to **Not Configured**.
- 5. In the message that appears, click OK.

References

For more information about how to configure a data synchronization task, see Overview of data synchronization scenarios.

5.Precheck troubleshooting5.1. Troubleshoot precheck failures

Before DTS runs a data synchronization task, DTS performs a precheck on the source and destination databases. This topic describes how to troubleshoot precheck failures based on the causes.

| Precheck item | Cause of failure | Solution |
|---|---|---|
| | The database account or password is invalid. | Obtain the valid password, and change the password of the source or destination database in the DTS console. The password is specified when you configure the data synchronization task. For more information, see Change the password of a database account. |
| Source database connectivity Destination database connectivity | The IP address that is used to access the source or destination database is disallowed to access the database. | For a MySQL database, authorize the database account again. The CIDR blocks of DTS servers must be added to the IP whitelist. For more information about the CIDR blocks of DTS servers in each region, see Add the CIDR blocks of DTS servers to the security settings of on-premises databases. For an SQL Server database, disable the firewall or trigger settings. For an Oracle database, set the TCP.VALIDNODE_CHECKING parameter to no and restart the process. |
| | A firewall is configured on the server where the source or destination database resides. | Disable the firewall settings. |
| | The network between DTS servers and the source or destination database is unavailable. | Contact Alibaba Cloud engineers for technical support. For more information, visit submit a ticket . |
| Source database version | DTS does not support the current database version. For more information, see Database types, initial synchronization types, and synchronization topologies. | Upgrade or downgrade your database version. |

| Precheck item | Cause of failure | Solution |
|-------------------------------------|--|--|
| Database existence | If the destination database does not exist in the destination instance, DTS may fail to create a database because of the following reasons: The database name contains special characters except letters, digits, underscores (_), and hyphens (-). The character set of the database is not UTF8, GBK, Latin1, or UTF8MB4. The account that is used to access the destination database does not have the required permissions. | Create a database in the destination instance or grant the required permissions to the database account. |
| Source database permissions | | |
| Destination database permissions | The account that is used to access the source or destination database does not have the required permissions. ? Note The permissions that are required for a database vary with the database type. For more information, see Overview of data synchronization scenarios. | Grant the required permissions to the database account. |
| | | |

| Precheck item | Cause of failure | Solution |
|---|--|--|
| Schema name conflict | The source and destination databases have the same name, or the source and destination tables have the same name. | Log on to the destination database, and change the database name or table names. Map the names of the conflicting objects to other objects in the destination database by using the object name mapping feature. For more information, see Specify the name of an object in the destination instance. Remove the conflicting objects. For more information, see Remove an object from a data synchronization task. |
| Value of server_id in the source database | The value of the server-id parameter is not set to an integer that is greater than or equal to 2. | Log on to the source database and change the value of the server-id parameter. For more information, see Value of server_id in a source database. |
| Whether binary logging is enabled for the source database | The binary logging feature is disabled for the source database. | Log on to the source database and enable the binary logging feature. For more information, see Source database binlogging. |
| Binary log format of the source database | The binary log format of the source database is set to ROW. | Log on to the source database, run the set global binlog_format='ROW'; command, and then restart the MySQL process. |
| Integrity of FOREIGN KEY constraint | The parent table on which a child table depends is not included in the required objects. This impairs the integrity of the FOREIGN KEY constraint. | Remove the foreign key dependencies from the child table that fails to pass the precheck. Add the parent table to the required objects. For more information, see Add an object to a data synchronization task. Remove the child table from the required objects. For more information, see Remove an object from a data synchronization task. |
| Storage engine | The storage engine of the source table is FEDERATED, MRG_MyISAM, or TokuDB. | Log on to the source database and set the storage engine of the source table to InnoDB. |

| Precheck item | Cause of failure | Solution |
|--------------------------------------|--|---|
| Character set | DTS does not support the character set of the required objects, such as UCS2. | Log on to the source database and set the character set of source tables to UTF8, GBK, Latin1, or UTF-8MB4. |
| Complicated topologies | DTS does not support the topology that you use for the source and destination instances. For more information, see Synchronization topologies. | Stop the conflicting task or wait for the task to complete, and then perform data synchronization again. |
| Format of MySQL database password | The format of the password that is used to access the source database is no longer valid. | Change the format of the database password. For more information, visit old_passwords. |

6.Synchronize data between MySQL databases

6.1. Configure two-way data synchronization between ApsaraDB RDS for MySQL instances

Data Transmission Service (DTS) supports real-time two-way data synchronization between two MySQL databases. This feature is applicable to scenarios such as active geo-redundancy (unit-based) and geo-disaster recovery. This topic describes how to configure two-way data synchronization between ApsaraDB RDS for MySQL instances.

Prerequisites

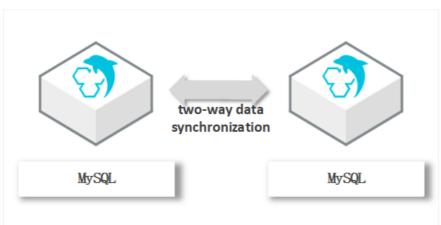
The source and destination ApsaraDB RDS for MySQL instances are created. For more information, see Create an RDS for MySQL instance.

Precautions

DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before you synchronize data, evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.

Supported synchronization topology

DTS supports two-way data synchronization only between two MySQL databases. DTS does not support two-way data synchronization between multiple MySQL databases.



Supported data sources

The following table describes the MySQL databases that are supported by two-way data synchronization. This topic uses ApsaraDB RDS for MySQL instances as the data sources. You can also follow the procedure to configure two-way data synchronization for other types of databases.

| Source database | Destination database |
|--|--|
| ApsaraDB RDS for MySQL instance User-created database hosted on ECS User-created database connected over Express Connect, VPN Gateway, or Smart Access Gateway | ApsaraDB RDS for MySQL instance User-created database hosted on ECS User-created database connected over Express Connect, VPN Gateway, or Smart Access Gateway |
| User-created database connected over a database gateway | User-created database connected over a database gateway |
| Self built database accessed through Cloud Enterprise Network(CEN) | Self built database accessed through Cloud Enterprise Network(CEN) |

Operations that can be synchronized

| Operati on type | SQL statements |
|--------------------|---|
| DML | INSERT, UPDATE, DELETE, and REPLACE |
| DDL | ALTER TABLE and ALTER VIEW CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW DROP INDEX and DROP TABLE RENAME TABLE TRUNCATE TABLE |

Conflict detection

To ensure data consistency, make sure that data records with the same primary key, business primary key, or unique key are updated only on one of the synchronization nodes. If data records are updated on both nodes, DTS responds to conflicts based on the conflict resolution policy that you have specified for the data synchronization task.

DTS checks and fixes conflicts to maximize the stability of two-way synchronization instances. DTS can detect the following types of conflicts:

• Uniqueness conflicts caused by INSERT operations

INSERT operations that do not comply with the uniqueness constraint cannot be synchronized. For example, if a record with the same primary key value is inserted into the two synchronization nodes at almost the same time, one of the inserted records fails to be synchronized. The synchronization fails because a record with the same primary key value already exists in the other node.

- Inconsistent records caused by UPDATE operations
 - If the records to be updated do not exist in the destination instance, DTS converts the UPDATE operation into an INSERT operation. However, uniqueness conflicts may occur.

- The primary keys or unique keys of the records to be inserted may conflict with those of existing records in the destination instance.
- Non-existent records to be deleted

The records to be deleted do not exist in the destination instance. In this case, DTS ignores the DELETE operation regardless of the conflict resolution policy that you have specified.

◯ Notice

- During two-way synchronization, the system time of the source and destination instances may be different. Synchronization latency may occur. For these reasons, DTS does not guarantee that the conflict detection mechanism can prevent all data conflicts. To perform two-way synchronization, make sure that data records with the same primary key, business primary key, or unique key are updated only on one of the synchronization nodes.
- DTS provides conflict resolution policies to prevent conflicts that may occur during data synchronization. You can select a conflict resolution policy when you configure a two-way data synchronization task.

Limits

• If you have selected one or more tables (not a database) for synchronization, do not use ghost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.

Notice To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.

Incompatibility with triggers

If the object to be synchronized is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source instance, the data in Table B becomes inconsistent between the source and destination instances.

To avoid this situation, before you synchronize data in Table B from the source instance, delete the trigger that is synchronized to the destination instance. For more information, see Configure synchronization when triggers exist.

• Limits on RENAME TABLE operations

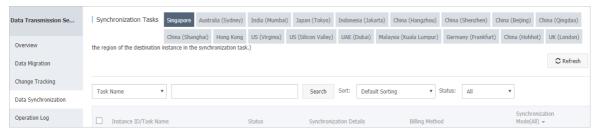
RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if a table to be synchronized is renamed during data synchronization, the data of this table is not synchronized to the destination database. To avoid this situation, you can select the database to which this table belongs as the object when you configure the data synchronization task.

• Limits on DDL synchronization direction

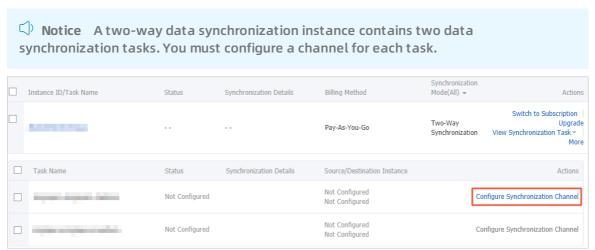
To ensure the stability of a two-way synchronization channel, you can synchronize DDL updates of a single table only in one direction. If DDL synchronization in a direction is configured, DDL synchronization in the opposite direction is not supported. Only DML operations can be synchronized in the opposite direction.

Procedure

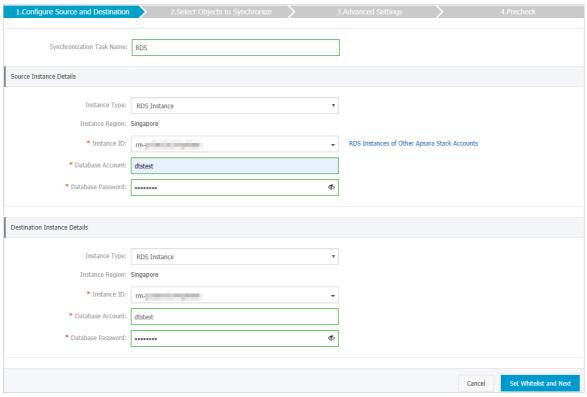
- 1. Purchase an instance for two-way data synchronization. For more information, see Purchase a data synchronization instance.
 - Notice On the buy page, set both Source Instance and Target Instance to MySQL and set Synchronization Topology to Two-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column of the first data synchronization task.



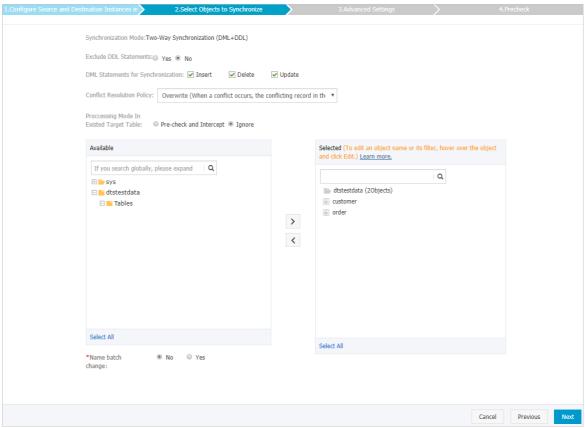
6. Configure the source and destination instances.



| Section | Parameter | Description |
|----------------------------|--|---|
| N/A | Synchronization Task Name DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. | |
| | Instance Type | Select RDS Instance. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the source RDS instance. |
| | | Enter the database account of the source RDS instance. |
| | Database Account | Notice If the database engine of the source RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password. |
| Source Instance Details | | |

| Section | Parameter | nrameter Description | |
|---------------------------------|-------------------|--|--|
| | Database Password | Enter the password for the source database account. | |
| | Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before you configure the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. | |
| | | Notice The Encryption parameter is available only for regions in mainland China and the Hong Kong (China) region. | |
| | Instance Type | Select RDS Instance. | |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. | |
| | Instance ID | Select the ID of the destination RDS instance. | |
| Destination Instance Details | Database Account | Enter the database account of the destination RDS instance. Notice If the database engine of the destination RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password. | |
| | Database Password | Enter the password for the destination database account. | |
| | Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before you configure the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. Notice The Encryption parameter is available only for regions in mainland China and the Hong Kong (China) region. | |

- 7. In the lower-right corner of the page, click **Set Whitelist and Next**.
- 8. Configure synchronization policy and objects.



| Section | Parameter | Description |
|---|---|--|
| | Exclude DDL Statements | To exclude DDL operations, select Yes. To include DDL operations, select No. |
| | | Notice After you select No, the table does not support synchronizing DDL operations in the opposite direction. |
| DML Statements for Synchronizati on | Select the type of DML operations to be synchronized. By default, the INSERT, UPDATE, and DELETE operations are selected. You can select the DML operation types based on your business requirements. | |

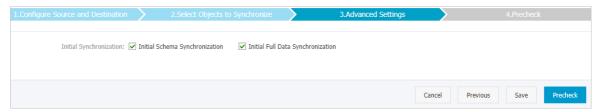
| Section | Parameter | Description |
|---------|----------------------------------|---|
| | Conflict Resolution Policy | Select the resolution policy in case of a synchronization conflict. By default, TaskFailed is selected. You can select a conflict resolution policy based on your business requirements. TaskFailed The default conflict resolution policy. If a conflict occurs during data synchronization, the synchronization task reports an error and exits the process. The task enters a failed state and you must manually resolve the conflict. Ignore |
| | | If a conflict occurs during data synchronization, the synchronization task ignores the current statement and continues the process. The conflicting records in the destination database are used. Overwrite |
| | | If a conflict occurs during data synchronization, the conflicting records in the destination database are overwritten. |

Synchronizati on policy

| Section | Parameter | Description |
|---------|----------------------------------|--|
| | | Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the source and destination databases do not contain identical table names, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. |
| | | Notice You can change the names of the tables to be synchronized by using the object name mapping feature. You can use this feature if the source and destination databases contain identical table names and tables in the destination database cannot be deleted or renamed. For more information, see Specify the name of an object in the destination instance. |
| | Processing Mode In Existed | Ignore: skips the precheck for identical table names in the source and destination databases. |
| | Target Table | Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. |
| | | |
| | | If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. |
| | | |

| Section | Parameter | Description |
|----------------------------|-----------|--|
| Objects to be synchronized | N/A | Select objects (tables or a database) from the Available section and click the icon to move the objects to the Selected section. Notice If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. If you select a table as the object to be synchronized, only schema changes in this table are synchronized to the destination database. |
| | | After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the names of the objects that are synchronized to the destination database by using the object name mapping feature. For more information about how to use this feature, see Specify the name of an object in the destination instance. |

- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.



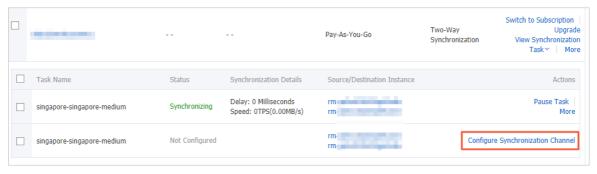
During initial synchronization, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization. Initial synchronization includes initial schema synchronization and initial full data synchronization. You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases.

Notice If tables to be synchronized in one direction are also included in the objects to be synchronized in the opposite direction, DTS does not synchronize these tables during initial synchronization.

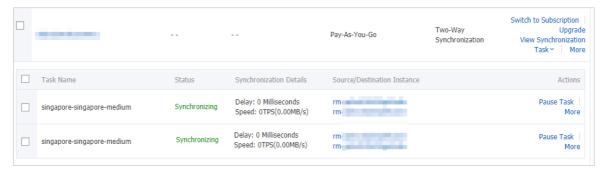
11. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed. Then, the data synchronization task starts.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.
 - You can view the status of the data synchronization task on the **Synchronization Tasks** page.
- 14. Find the second data synchronization task and click **Configure Synchronization Channel** in the Actions column. Configure data synchronization by following step 5 to step 12.



15. After the second data synchronization task is configured, wait until both tasks are in the **Synchronizing** state. The two-way data synchronization task is configured.



6.2. Configure one-way data synchronization between ApsaraDB RDS for MySQL instances

Data Transmission Service (DTS) supports data synchronization between two MySQL databases. This topic describes how to configure one-way data synchronization between two ApsaraDB RDS for MySQL instances.

Prerequisites

- The source and destination ApsaraDB RDS for MySQL instances for data synchronization are created. For more information, see Create an RDS instance.
- The databases in the source and destination RDS instances are MySQL databases.

Notes

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- If you have selected one or more tables (not a database) for synchronization, do not use ghost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.

Notice To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.

- You cannot synchronize data between ApsaraDB RDS for MySQL instances that reside in Zone A of the China (Hong Kong) region.
- The source and destination ApsaraDB RDS for MySQL instances must have internal endpoints.
- If the source database does not have primary keys or UNIQUE constraints, and fields are not required to be unique, duplicate data may exist in the destination database.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After initial full data synchronization, the tablespace of the destination instance is larger than that of the source instance.

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization
- One-way many-to-one synchronization
- Two-way one-to-one synchronization

For more information about synchronization topologies, see Synchronization topologies.

SQL operations that can be synchronized

| Operati on type | SQL statements |
|--------------------|---|
| DML | INSERT, UPDATE, DELETE, and REPLACE |
| DDL | ALTER TABLE and ALTER VIEW CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW DROP INDEX and DROP TABLE RENAME TABLE TRUNCATE TABLE |

Limits

• Incompatibility of triggers

If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source instance, the data in Table B becomes inconsistent between the source and destination instances.

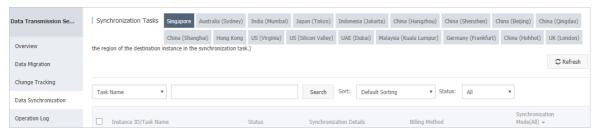
To avoid this situation, you must delete the trigger that is synchronized to the destination instance and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.

• Limits on RENAME TABLE operations

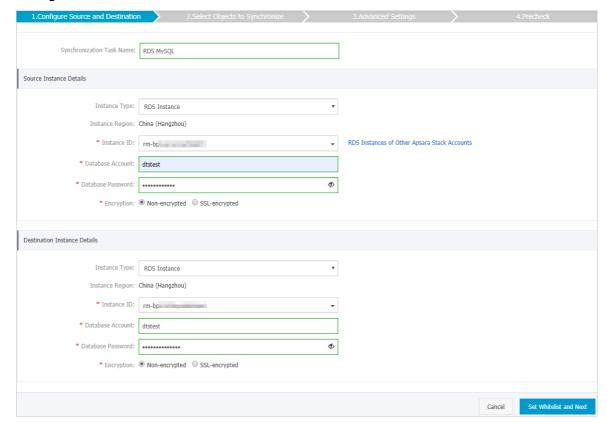
RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

Procedure

- 1. Purchase an instance for one-way data synchronization. For more information, see Purchase a data synchronization instance.
 - Note Select MySQL for both the source instance and the destination instance and select One-Way Synchronization as the synchronization topology.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

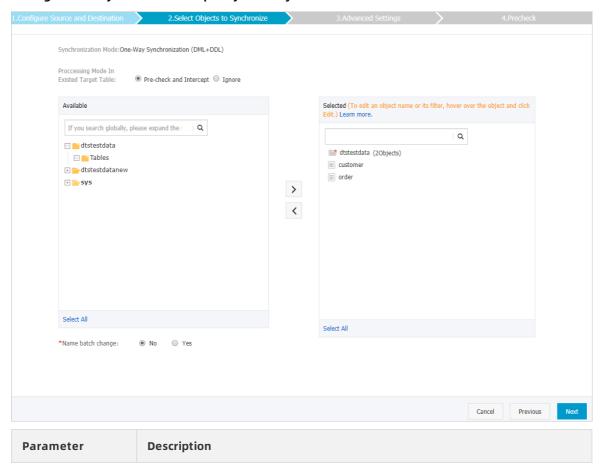


| Section | Parameter | Description |
|---------|-------------------------------|---|
| N/A | Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select RDS Instance. |
| | Instance Region | The region of the source instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the source RDS instance. |
| | | |

| Section | Parameter | Description |
|------------------------------------|----------------------|---|
| Source Instance Details | Database Account | Once If the database engine of the source RDS instance. Note If the database engine of the source RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password. |
| | Database Password | Enter the password for the database account. |
| | Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. 7 Note The Encryption parameter is available only in mainland China and Hong Kong(China). |
| | Instance Type | Select RDS Instance. |
| | Instance Region | The region of the destination instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the destination RDS instance. |
| | Database Account | Onte If the database engine of the destination RDS instance. Note If the database engine of the destination RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password. |
| | Database Password | Enter the password for the database account. |
| Destination Instance Details | | |

| Section | Parameter | Description |
|------------|---|---|
| Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. | |
| | | ? Note The Encryption parameter is available only in mainland China and Hong Kong(China). |

- 7. In the lower-right corner of the page, click **Set Whitelist and Next**.
- 8. Configure the synchronization policy and objects.

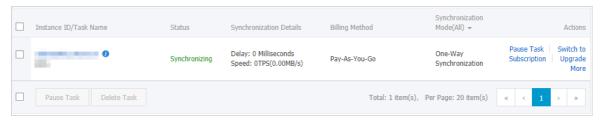


| Parameter | Description | | |
|--------------------------------------|---|--|--|
| | Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. | | |
| | Note If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. | | |
| Processing Mode In Existed Target | Ignore: skips the precheck for identical table names in the source and destination databases. | | |
| Table | Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. | | |
| | If the source and destination databases have the same schema, and the primary key of a record in the destination database is the same as that in the source database, the record remains unchanged during initial data synchronization. However, the record is overwritten during incremental data synchronization. | | |
| | If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. | | |
| | | | |
| | Select objects from the Available section and click the > icon to move the | | |
| | objects to the Selected section. | | |
| | You can select tables and databases as the objects to be synchronized. | | |
| | ? Note | | |
| Objects to be synchronized | If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. | | |
| | After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination instance by using the object name mapping feature provided by DTS. For more information about how to use this feature, see Specify the name of an object in the destination instance. | | |
| | | | |

- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.



- 11. In the lower-right corner of the page, click Precheck.
 - Note
 Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
 If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.



6.3. Synchronize data from a user-created MySQL database hosted on ECS to an ApsaraDB RDS for MySQL instance

This topic describes how to synchronize data from a user-created MySQL database to an ApsaraDB RDS for MySQL instance by using Data Transmission Service (DTS).

Prerequisites

- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The destination RDS instance is created. For more information, see Create an RDS instance.

Limits

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- If you have selected one or more tables (not a database) for synchronization, do not use ghost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.

Notice To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.

- You cannot synchronize data to an ApsaraDB RDS for MySQL instance that resides in Zone A of the China (Hong Kong) region.
- The destination ApsaraDB RDS for MySQL instance must have an internal endpoint.
- If the source database does not have primary keys or UNIQUE constraints, and fields are not required to be unique, duplicate data may exist in the destination database.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After initial full data synchronization, the tablespace of the destination instance is larger than that of the source instance.

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way many-to-one synchronization
- One-way cascade synchronization
- Two-way one-to-one synchronization

Note For more information about two-way synchronization, see Configure two-way data synchronization between ApsaraDB RDS for MySQL instances.

SQL operations that can be synchronized

| Operati on type | SQL statements |
|--------------------|---|
| DML | INSERT, UPDATE, DELETE, and REPLACE |
| DDL | ALTER TABLE and ALTER VIEW CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW DROP INDEX and DROP TABLE RENAME TABLE TRUNCATE TABLE |

Limits

• Incompatibility of triggers

If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source instance, the data in Table B becomes inconsistent between the source and destination instances.

To avoid this situation, you must delete the trigger that is synchronized to the destination instance and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.

• Limits on RENAME TABLE operations

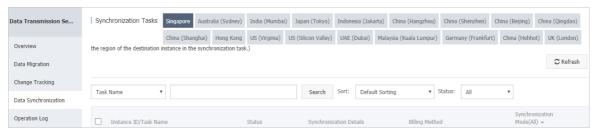
RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

Preparations

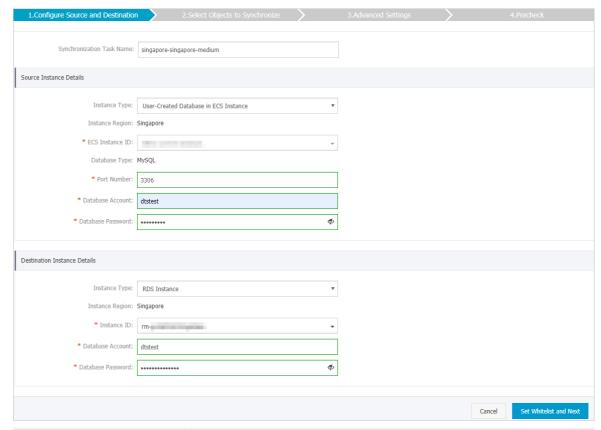
Before configuring the data synchronization task, you must create a database account and configure binary logging. For more information, see Create an account for a user-created MySQL database and configure binary logging.

Procedure

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note Select MySQL for both the source instance and the destination instance.
 Select One-Way Synchronization as the synchronization topology.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click **Data Synchronization**.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



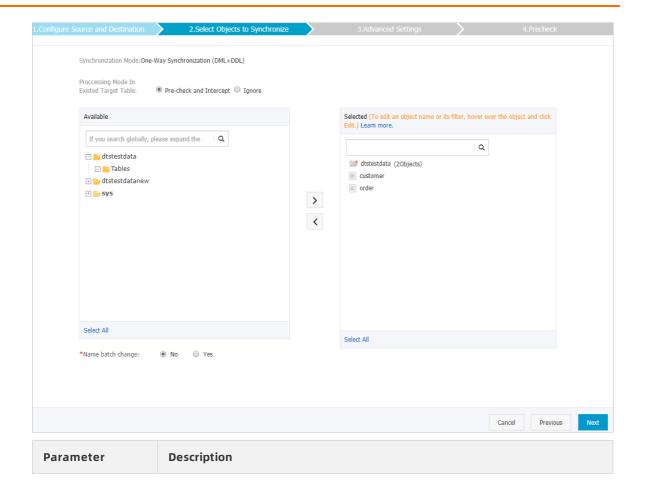
- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



| Section | Parameter | Description |
|---------|----------------------------------|--|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select User-Created Database in ECS Instance. |
| | Instance Region | The region of the source instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | ECS Instance ID | Select the ID of the ECS instance that is connected to the user-created MySQL database. |
| | | |

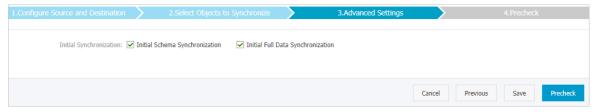
| Section Source | Parameter | Description |
|-------------------------------------|----------------------|---|
| Instance Details | Database Type | The value of this parameter is set to MySQL and cannot be changed. |
| | Port Number | Enter the service port number of the user-created MySQL database. The default port number is 3306. |
| | Database Account | Enter the account for the user-created MySQL database. The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. |
| | Database Password | Enter the password for the account of the user-created MySQL database. |
| | Instance Type | Select RDS Instance. |
| | Instance Region | The region of the destination instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the destination RDS instance. |
| Destinatio n Instance Details | Database Account | ? Note If the database engine of the destination RDS instance. instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password. |
| | Database Password | Enter the password for the database account. |
| | Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. |
| | | ? Note The Encryption parameter is available only in mainland China and Hong Kong(China). |
| | | |

- 7. In the lower-right corner of the page, click **Set Whitelist and Next**.
- 8. Configure the synchronization policy and objects.

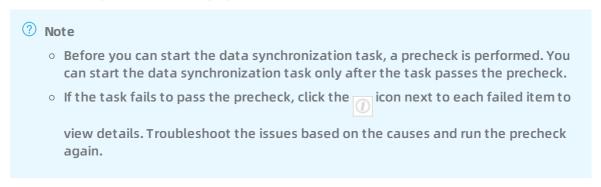


| Parameter | Description |
|-------------------------------|---|
| | Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. |
| | ? Note If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. |
| Processing Mode | Ignore: skips the precheck for identical table names in the source and destination databases. |
| In Existed Target Table | Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. If the source and destination databases have the same schema, and the primary key of a record in the destination database is the same as that in the source database, the record remains unchanged during initial data synchronization. However, the record is overwritten during incremental data synchronization. If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. |
| | Select objects from the Available section and click the > icon to move the |
| | objects to the Selected section. |
| Objects to be synchronized | Note If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination instance by using the object name mapping feature provided by DTS. For more information about how to use this feature, see Specify the name of an object in the destination instance. |

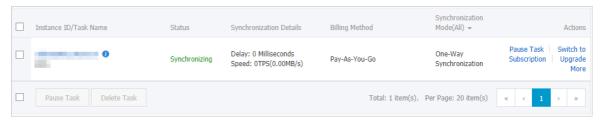
- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.



- During initial synchronization, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization.
- Initial synchronization includes initial schema synchronization and initial full data synchronization. You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases.
- 11. In the lower-right corner of the page, click Precheck.



- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the synchronization task is in the **Synchronizing** state.



6.4. Synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an ApsaraDB RDS for MySQL instance

This topic describes how to synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an ApsaraDB RDS for MySQL instance by using Data Transmission Service (DTS).

Prerequisites

- The destination RDS instance is created. For more information, see Create an RDS instance.
- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The user-created MySQL database is connected to Alibaba Cloud VPC over Express Connect, VPN Gateway, or Smart Access Gateway. For more information, see Connect to local IDCs.

Note DTS is allowed to access the VPC to which the user-created MySQL database belongs. For more information, see Configure a route between DTS and Express Connect, VPN Gateway, or Smart Access Gateway.

Notes

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- If you have selected one or more tables (not a database) for synchronization, do not use ghost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.

Notice To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.

- You cannot synchronize data to an ApsaraDB RDS for MySQL instance that resides in Zone A of the China (Hong Kong) region.
- The destination ApsaraDB RDS for MySQL instance must have an internal endpoint.
- If the source database does not have primary keys or UNIQUE constraints, and fields are not required to be unique, duplicate data may exist in the destination database.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After initial full data synchronization, the tablespace of the destination instance is larger than that of the source instance.

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization

- One-way many-to-one synchronization
- Two-way one-to-one synchronization

For more information about synchronization topologies, see Synchronization topologies.

SQL operations that can be synchronized

| Operati on type | SQL statements |
|--------------------|---|
| DML | INSERT, UPDATE, DELETE, and REPLACE |
| DDL | ALTER TABLE and ALTER VIEW CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW DROP INDEX and DROP TABLE RENAME TABLE TRUNCATE TABLE |

Limits

• Incompatibility of triggers

If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source instance, the data in Table B becomes inconsistent between the source and destination instances.

To avoid this situation, you must delete the trigger that is synchronized to the destination instance and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.

• Limits on RENAME TABLE operations

RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

Preparations

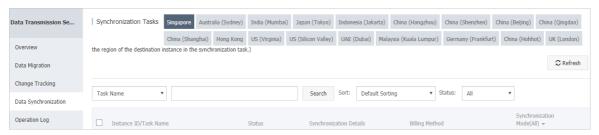
Before configuring the data synchronization task, you must create a database account and configure binary logging. For more information, see Create an account for a user-created MySQL database and configure binary logging.

Procedure

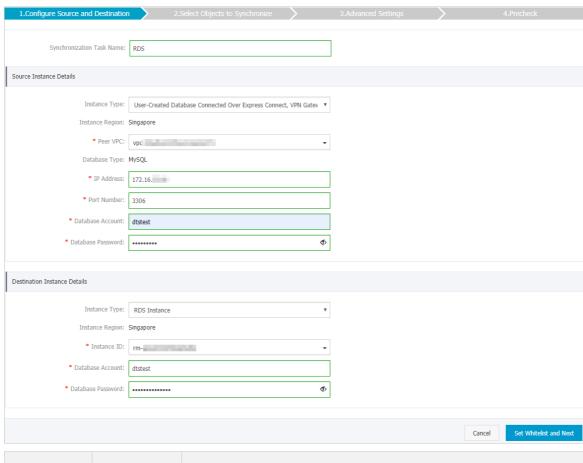
1. Purchase a data synchronization instance. For more information, see Purchase procedure.

Note Select MySQL for both the source instance and the destination instance.
Select One-Way Synchronization as the synchronization topology.

- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

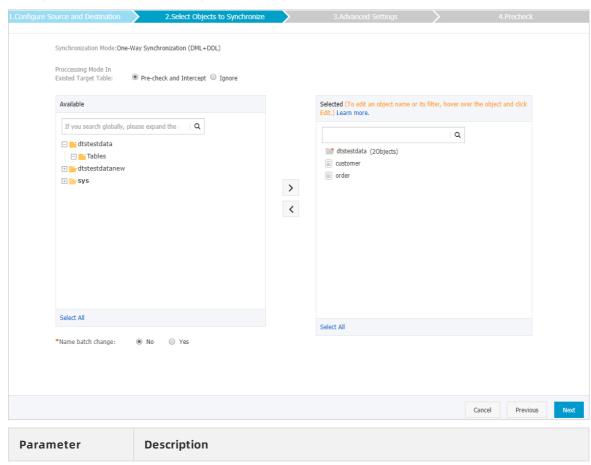


| Section | Parameter | Description |
|---------|----------------------------------|--|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway. |

| Section | Parameter | Description |
|-------------------------------------|----------------------|--|
| | Instance Region | The region of the source instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| Source | Peer VPC | Select the ID of the VPC that is connected to the user-created MySQL database. |
| Instance Details | Database Type | The value of this parameter is set to MySQL and cannot be changed. |
| | IP Address | Enter the server IP address of the user-created MySQL database. |
| | Port Number | Enter the service port number of the user-created MySQL database. The default port number is 3306. |
| | Database Account | Enter the account for the user-created MySQL database. The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. |
| | Database Password | Enter the password for the database account. |
| | Instance Type | Select RDS Instance. |
| | Instance Region | The region of the destination instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the destination RDS instance. |
| | | Enter the database account for the destination RDS instance. |
| Destinatio n Instance Details | Database Account | Note If the database engine of the destination RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password. |
| | Database Password | Enter the password for the database account. |
| | | |

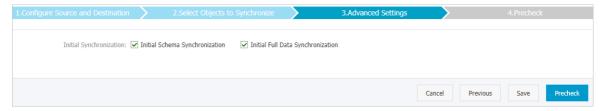
| Section | Parameter | Description |
|---------|------------|---|
| | Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. |
| | | Note The Encryption parameter is available only in mainland China and Hong Kong(China). |

- 7. In the lower-right corner of the page, click Set Whitelist and Next.
- 8. Configure the synchronization policy and objects.

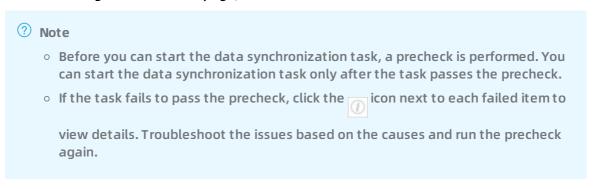


| Parameter | Description | |
|---|---|--|
| Processing Mode In Existed Target Table | Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. Note If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. Ignore: skips the precheck for identical table names in the source and destination databases. Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. If the source and destination databases have the same schema, and the primary key of a record in the destination database is the same as that in the source database, the record remains unchanged during initial data synchronization. However, the record is overwritten during incremental data synchronization. If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. | |
| Objects to be synchronized | Select objects from the Available section and click the icon to move the objects to the Selected section. You can select tables and databases as the objects to be synchronized. Note If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination instance by using the object name mapping feature provided by DTS. For more information about how to use this feature, see Specify the name of an object in the destination instance. | |

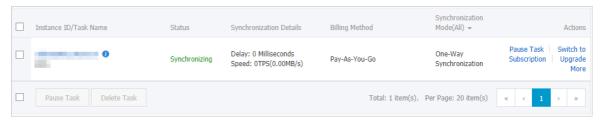
- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.



- During initial synchronization, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization.
- Initial synchronization includes initial schema synchronization and initial full data synchronization. You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases.
- 11. In the lower-right corner of the page, click Precheck.



- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.



6.5. Synchronize data from an ApsaraDB RDS for MySQL instance to a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway

This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway by using Data Transmission Service (DTS).

Prerequisites

- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
 - Note We recommend that you make sure the version of the source and destination MySQL databases is the same.
- The user-created MySQL database is connected to Alibaba Cloud VPC over Express Connect, VPN Gateway, or Smart Access Gateway. For more information, see Connect to local IDCs.
 - Note DTS is allowed to access the VPC to which the user-created MySQL database belongs. For more information, see Configure a route between DTS and Express Connect, VPN Gateway, or Smart Access Gateway.

Notes

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- You cannot synchronize data from an ApsaraDB RDS for MySQL instance that resides in Zone A of the China (Hong Kong) region.
- The source ApsaraDB RDS for MySQL instance must have an internal endpoint.
- During initial full data synchronization, concurrent INSERT operations cause segments in the tables of the destination instance. After initial full data synchronization, the tablespace of the destination instance is larger than that of the source instance.

Limits

- If you have selected one or more tables (not a database) for synchronization, do not use ghost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.
 - Notice To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.
- If the source database does not have primary keys or UNIQUE constraints, and fields are not required to be unique, duplicate data may exist in the destination database.

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization
- One-way many-to-one synchronization
- Two-way one-to-one synchronization

For more information about synchronization topologies, see Synchronization topologies.

SQL operations that can be synchronized

| Operati on type | SQL statements |
|--------------------|---|
| DML | INSERT, UPDATE, DELETE, and REPLACE |
| DDL | ALTER TABLE and ALTER VIEW CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW DROP INDEX and DROP TABLE RENAME TABLE TRUNCATE TABLE |

Limits

• Incompatibility of triggers

If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source instance, the data in Table B becomes inconsistent between the source and destination instances.

To avoid this situation, you must delete the trigger that is synchronized to the destination instance and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.

• Limits on RENAME TABLE operations

RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

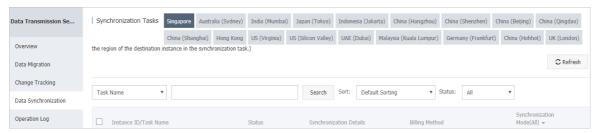
Procedure

1. Create a data synchronization task. For more information, see Purchase procedure.

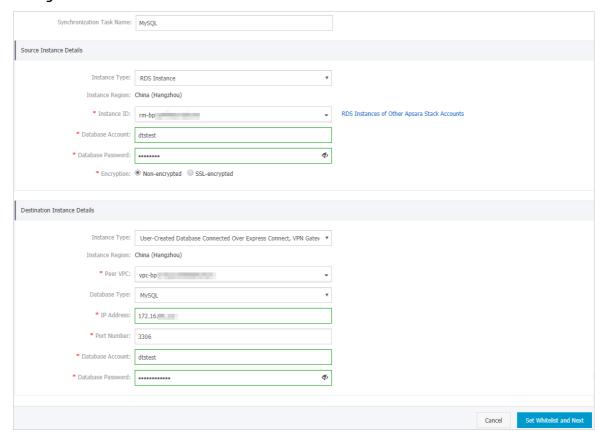
Note Select MySQL for both the source instance and the destination instance.
Select One-Way Synchronization as the synchronization topology.

2. Log on to the DTS console.

- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



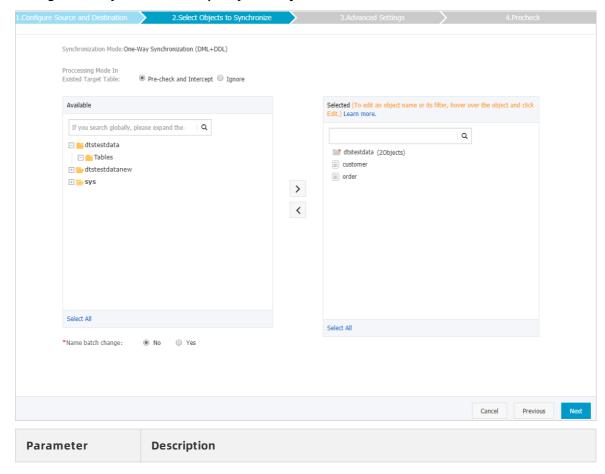
- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



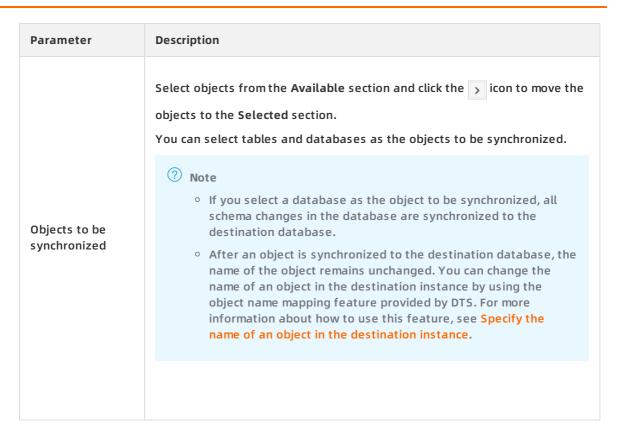
| Section | Parameter | Description |
|---------|----------------------------------|--|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select RDS Instance. |
| | Instance Region | The region of the source instance that you select when purchasing the data synchronization instance. You cannot change the value of this parameter. |

| Section | Parameter | Description |
|-------------------------------|----------------------|---|
| | Instance ID | Select the ID of the source RDS instance. |
| Source Instance Details | Database Account | Onte If the database type of the source RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password. |
| | Database Password | Enter the password for the database account. |
| | Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. |
| | | ? Note The Encryption parameter is available only in mainland China and Hong Kong(China). |
| | Instance Type | Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway. |
| | Instance Region | The region of the destination instance that you select when purchasing the data synchronization instance. You cannot change the value of this parameter. |
| | Peer VPC | Select the ID of the VPC that is connected to the user-created MySQL database. |
| | Database Type | This parameter is set to MySQL and cannot be modified. |
| Destinatio n Instance | IP Address | Enter the IP address of the user-created MySQL database. |
| Details | Port Number | Enter the service port number of the user-created MySQL database. The default port number is 3306. |
| | Database Account | Once The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. |
| | Database Password | Enter the password for the database account. |

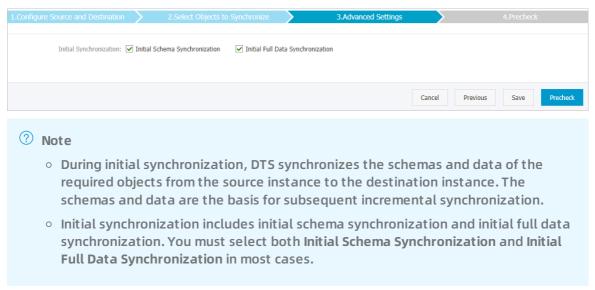
- 7. In the lower-right corner of the page, click Set Whitelist and Next.
- 8. Configure the synchronization policy and objects.



| Parameter | Description | |
|--------------------------------------|---|--|
| | Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. | |
| | Note If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. | |
| Processing Mode In Existed Target | Ignore: skips the precheck for identical table names in the source and destination databases. | |
| Table | Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. | |
| | If the source and destination databases have the same schema, and the primary key of a record in the destination database is the same as that in the source database, the record remains unchanged during initial data synchronization. However, the record is overwritten during incremental data synchronization. | |
| | If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. | |
| | | |



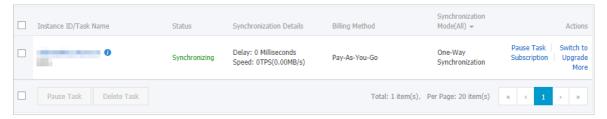
- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.



11. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.



6.6. Synchronize data between ApsaraDB RDS for MySQL instances that belong to different Alibaba Cloud accounts

This topic describes how to synchronize data between ApsaraDB RDS for MySQL instances that belong to different Alibaba Cloud accounts by using Data Transmission Service (DTS).

Prerequisites

- The source and destination ApsaraDB RDS for MySQL instances for data synchronization are created. For more information, see Create an RDS instance.
- The databases in the source and destination RDS instances are MySQL databases.
- The source and destination ApsaraDB RDS for MySQL instances must have internal endpoints.

Limits

• DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.

• If you have selected one or more tables (not a database) for synchronization, do not use ghost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.

Notice To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.

- You cannot synchronize data between ApsaraDB RDS for MySQL instances that reside in Zone A of the China (Hong Kong) region.
- If the source database does not have primary keys or UNIQUE constraints, and fields are not required to be unique, duplicate data may exist in the destination database.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After initial full data synchronization, the tablespace of the destination instance is larger than that of the source instance.

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization
- One-way many-to-one synchronization

For more information about synchronization topologies, see Synchronization topologies.

SQL operations that can be synchronized

| Operati on type | SQL statements |
|--------------------|---|
| DML | INSERT, UPDATE, DELETE, and REPLACE |
| DDL | ALTER TABLE and ALTER VIEW CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW DROP INDEX and DROP TABLE RENAME TABLE TRUNCATE TABLE |

Limits

• Incompatibility of triggers

If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source instance, the data in Table B becomes inconsistent between the source and destination instances.

To avoid this situation, you must delete the trigger that is synchronized to the destination instance and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.

• Limits on RENAME TABLE operations

RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

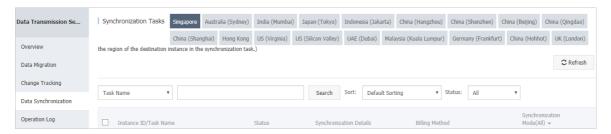
Preparations

Set the Alibaba Cloud account that owns the destination RDS instance as a trusted account. This allows DTS to access the cloud resources of the Alibaba Cloud account that owns the source RDS instance. For more information, see Configure RAM authorization for cross-account data migration and synchronization.

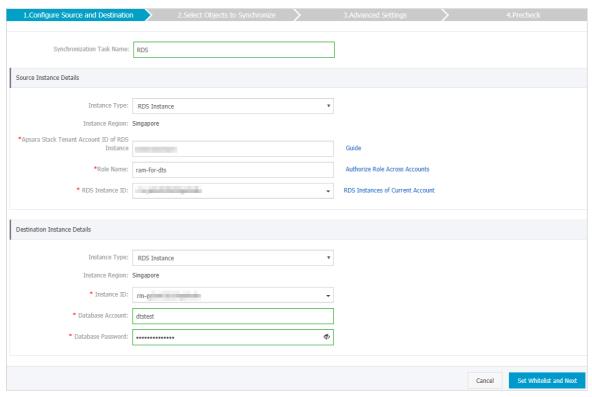
Note To authorize the Alibaba Cloud account that owns the destination instance, you must log on to the RAM console with the Alibaba Cloud account that owns the source instance. Then, you can create a data migration task or data synchronization task by using the Alibaba Cloud account that owns the destination instance.

Procedure

- Purchase a data synchronization instance by using the Alibaba Cloud account that owns the destination RDS instance. For more information, see <u>Purchase a data synchronization</u> instance.
 - Note Select MySQL for both the source instance and the destination instance.
 Select One-Way Synchronization as the synchronization topology.
- 2. Log on to the DTS console with the Alibaba Cloud account that owns the destination RDS instance.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



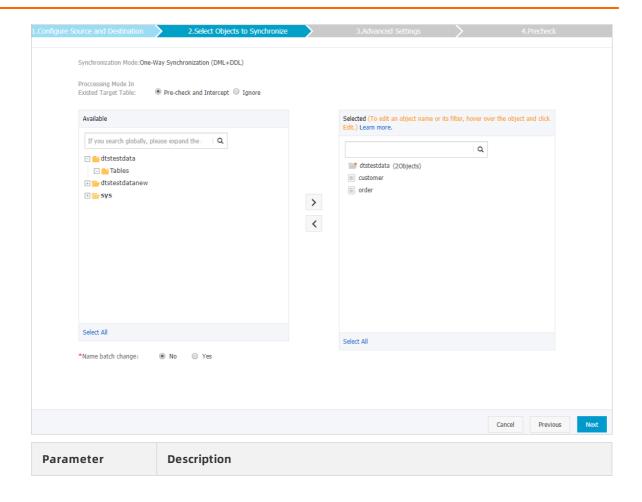
- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



| Section | Parameter | Description |
|-------------------------------|---|---|
| N/A | Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select RDS Instance. |
| Source Instance Details | Instance Region | The region of the source instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Apsara Stack Tenant Account ID of RDS Instance | Enter the ID of the Alibaba Cloud account that owns the source RDS instance. ? Note Before you configure this parameter, click RDS Instances of Other Apsara Stack Accounts in the Source Instance Details section. Instance Type: RDS Instance |
| | Role Name | Enter the name of the RAM role that you configured earlier in Preparations. |
| | RDS Instance ID | Select the ID of the source RDS instance. |

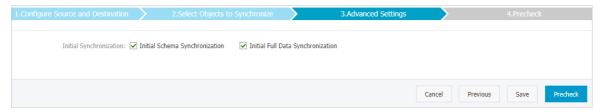
| Section | Parameter | Description |
|------------------------------------|----------------------|---|
| | | |
| | Instance Type | Select RDS Instance. |
| | Instance Region | The region of the destination instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the destination RDS instance. |
| Destination Instance Details | Database Account | Once If the database engine of the destination RDS instance. Note If the database engine of the destination RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password. |
| | Database Password | Enter the password for the database account. |
| | Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. |
| | | ? Note The Encryption parameter is available only in mainland China and Hong Kong(China). |

- 7. In the lower-right corner of the page, click **Set Whitelist and Next**.
- 8. Configure the synchronization policy and objects.



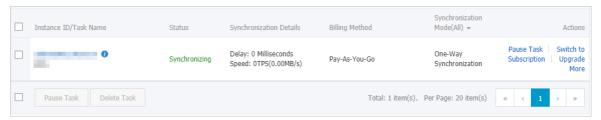
| Parameter | Description | | |
|-------------------------------|---|--|--|
| | Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the da synchronization task cannot be started. | | |
| | Note If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. | | |
| Processing Mode | Ignore: skips the precheck for identical table names in the source and destination databases. | | |
| In Existed Target Table | Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. If the source and destination databases have the same schema, and the primary key of a record in the destination database is the same as that in the source database, the record remains unchanged during initial data synchronization. However, the record is overwritten during incremental data synchronization. If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. | | |
| | Select objects from the Available section and click the icon to move the | | |
| | objects to the Selected section. | | |
| Objects to be synchronized | Note If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination instance by using the object name mapping feature provided by DTS. For more information about how to use this feature, see Specify the name of an object in the destination instance. | | |

- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.



- During initial synchronization, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization.
- Initial synchronization includes initial schema synchronization and initial full data synchronization. You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases.
- 11. In the lower-right corner of the page, click Precheck.
 - Note
 Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
 If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

You can view the status of the data synchronization task on the Data Synchronization page.



7.Synchronize data from MySQL databases to other databases

7.1. Synchronize data from an ApsaraDB RDS for MySQL instance to an Apsara PolarDB for MySQL cluster

Apsara PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is a high-performance, high-availability, easy-to-use, and reliable service that is compatible with the MySQL database engine. This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to an Apsara PolarDB for MySQL cluster by using Data Transmission Service (DTS).

Prerequisites

An Apsara PolarDB for MySQL cluster is created. For more information, see Create an Apsara PolarDB for MySQL cluster.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- If you have selected one or more tables (not a database) for synchronization, do not use ghost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.
 - Notice To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source instance.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination database.

SQL operations that can be synchronized

| Operati on type | SQL statements |
|--------------------|---|
| DML | INSERT, UPDATE, DELETE, and REPLACE |
| DDL | ALTER TABLE and ALTER VIEW CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW DROP INDEX and DROP TABLE RENAME TABLE TRUNCATE TABLE |

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization
- One-way many-to-one synchronization

For more information about synchronization topologies, see Synchronization topologies.

Limits

• Incompatibility with triggers

If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source database, the data in Table B becomes inconsistent between the source and destination databases.

To avoid this situation, you must delete the trigger that is synchronized to the destination database and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.

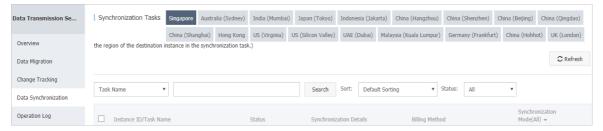
• Limits on RENAME TABLE operations

RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

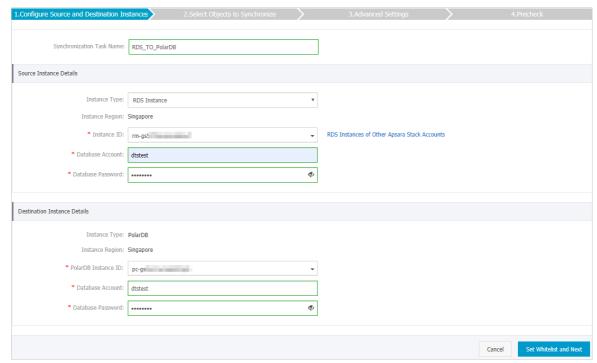
Procedure

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to MySQL, Target Instance to PolarDB, and Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.

4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

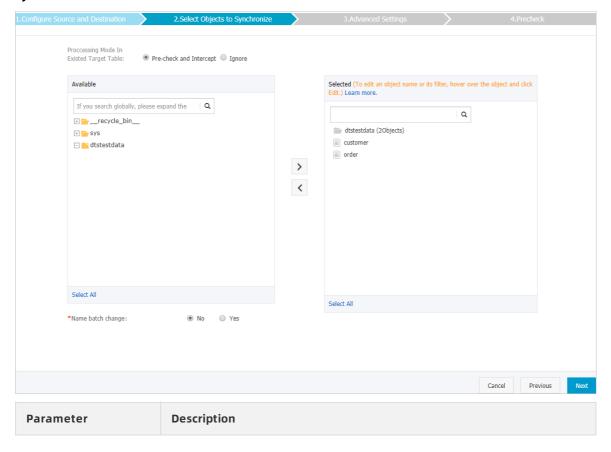


| Section | Parameter | Description |
|---------|----------------------------------|---|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select RDS Instance. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | | |

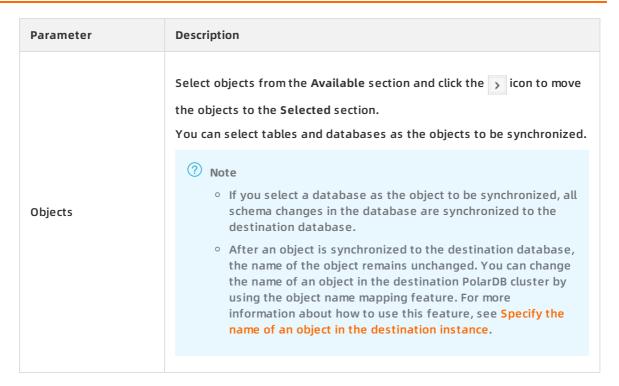
| Section | Parameter | Description |
|-------------------------------|------------------------|---|
| Source Instance Details | Database Account | Enter the database account of the source RDS instance. |
| | | Note The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. If the database engine of the source RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password. |
| | Database Password | Enter the password for the source database account. |
| | Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. Note The Encryption parameter is available only in mainland China and Hong Kong(China). |
| | Instance Type | The value of this parameter is set to PolarDB and cannot be changed. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| Destination Instance | PolarDB Instance ID | Select the ID of the destination PolarDB cluster. |
| Details | Database Account | Enter the database account of the destination PolarDB cluster. Note The database account must have the ALL permission for the objects to be synchronized. |
| | Database Password | Enter the password for the destination database account. |

7. In the lower-right corner of the page, click **Set Whitelist and Next**.

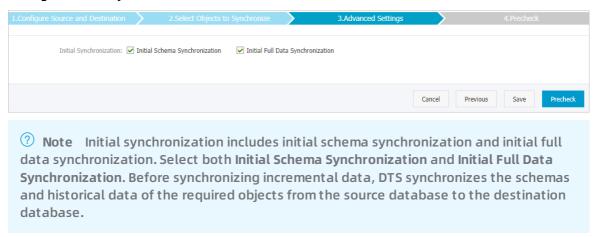
- Note The CIDR blocks of DTS servers are automatically added to the whitelist of the source RDS instance and the destination PolarDB cluster. This ensures that DTS servers can connect to the source RDS instance and the destination PolarDB cluster.
- 8. Configure the processing mode in existing destination tables and the objects to be synchronized.



| Parameter | Description | |
|--|--|--|
| | Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. | |
| | Note If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. | |
| | Ignore: skips the precheck for identical table names in the source and destination databases. | |
| Processing Mode In Existed Target Table | Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. | |
| | | |



- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.



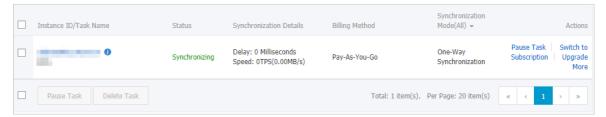
11. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the

Synchronizing state.

On the Synchronization Tasks page, view the status of the data synchronization task.



7.2. Synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for MySQL cluster

AnalyticDB for MySQL is a real-time online analytical processing (RT-OLAP) service developed by Alibaba Cloud for online data analysis with high concurrency. AnalyticDB for MySQL can analyze petabytes of data from multiple dimensions at millisecond-level timing to provide you with data-driven insights into your business. This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for MySQL cluster by using Data Transmission Service (DTS). AnalyticDB for MySQL allows you to build internal business intelligence (BI) systems, interactive query systems, and real-time report systems.

Prerequisites

- The tables to be synchronized from the ApsaraDB RDS for MySQL instance contain primary keys.
- An AnalyticDB for MySQL cluster is created. For more information, see Create an AnalyticDB for MySQL cluster.
- The destination AnalyticDB for MySQL cluster has sufficient storage space.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on objects during data synchronization. Otherwise, data synchronization may fail.
- If the disk space usage of nodes in an AnalyticDB for MySQL cluster reaches 80%, the cluster is locked. We recommend that you estimate the required disk space based on the objects to be synchronized. You must ensure that the destination cluster has sufficient storage space.

Supported source database types

You can use DTS to synchronize data from the following types of MySQL databases:

- ApsaraDB RDS for MySQL
- User-created database hosted on ECS
- User-created database connected over Express Connect, VPN Gateway, or Smart Access Gateway

This topic uses **ApsaraDB RDS** for **MySQL** as an example to describe how to configure a data synchronization task. You can also follow the procedure to configure data synchronization tasks for user-created MySQL databases.

Note If your source database is a user-created MySQL database, you must prepare the environments that are required for the source database. For more information, see Preparation overview.

SQL operations that can be synchronized

- DDL operations: CREATE TABLE, DROP TABLE, RENAME TABLE, TRUNCATE TABLE, ADD COLUMN, and DROP COLUMN
- DML operations: INSERT, UPDATE, and DELETE

Note If the data type of a field in the source table is changed during data synchronization, an error message is generated and the data synchronization task stops. You can submit a ticket or manually troubleshoot the issue. For more information, see Troubleshoot the synchronization failure that occurs due to field type changes.

Permissions required for database accounts

| Database | Required permission |
|------------------------|--|
| ApsaraDB RDS for MySQL | The REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects |
| AnalyticDB for MySQL | The read/write permissions on the objects to be synchronized |

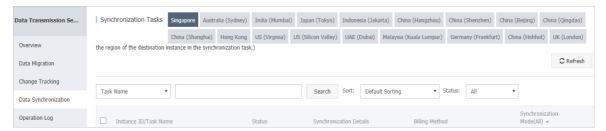
Data type mappings

The data types of ApsaraDB RDS for MySQL and AnalyticDB for MySQL do not have one-to-one correspondence. During initial schema synchronization, DTS converts the data types of the source database into those of the destination database. For more information, see Data type mappings for initial schema synchronization.

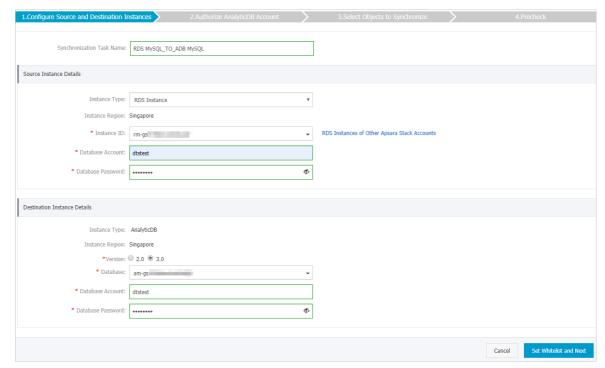
Procedure

1. Purchase a data synchronization instance.

- Note On the buy page, set Source Instance to MySQL, set Target Instance to AnalyticDB for MySQL, and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

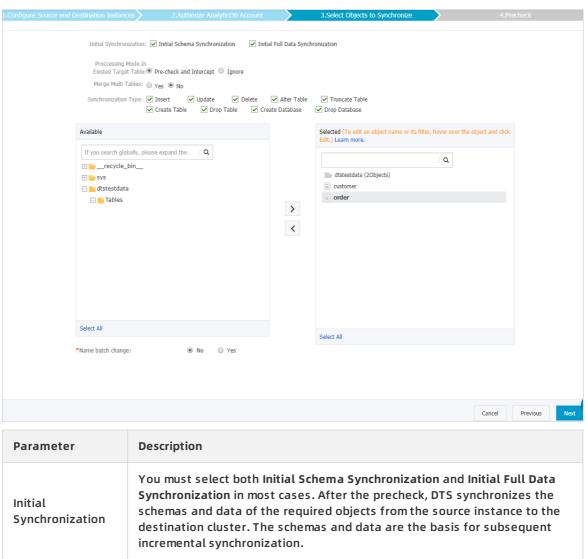


| Parameter | Description |
|-------------------------------|--|
| Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| Instance Type | Select RDS Instance. |
| | |
| | Synchronizati on Task Name |

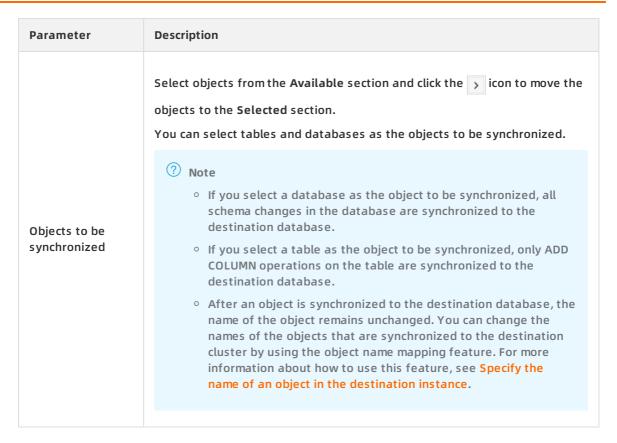
| Section | Parameter | Description |
|-------------------------------|----------------------|--|
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the source RDS instance. |
| Source Instance Details | Database Account | Enter the database account of the source RDS instance. For more information about the permissions that are required for the account, see Permissions required for database accounts. Onte If the database engine of the source RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password. |
| | Database Password | Enter the password for the source database account. |
| | Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before you configure the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. Notice The Encryption parameter is available only for regions in mainland China and the Hong Kong (China) region. |
| | Instance Type | The value of this parameter is set to AnalyticDB and cannot be changed. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| Destination | Version | Select 3.0. |
| Instance Details | Database | Select the ID of the destination AnalyticDB for MySQL cluster. |
| | Database Account | Enter the database account of the AnalyticDB for MySQL cluster. For more information about the permissions that are required for the account, see Permissions required for database accounts. |
| | Database Password | Enter the password for the destination database account. |

7. In the lower-right corner of the page, click **Set Whitelist and Next**.

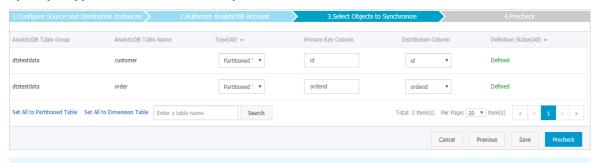
8. Configure the synchronization policy and objects.



| Parameter | Description | |
|---|--|--|
| Processing Mode In Existed Target Table | Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the source and destination databases do not contain identical table names, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. Note You can change the names of the tables to be synchronized by using the object name mapping feature. You can use this feature if the source and destination databases contain identical table names and tables in the destination database cannot be deleted or renamed. For more information, see Specify the name of an object in the destination instance. Ignore: skips the precheck for identical table names in the source and destination databases. Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. If the source and destination databases have the same schema, DTS does not synchronize data records that have the same primary keys as data records in the destination database. If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. | |
| Merge Multi Tables | If you select Yes, DTS adds thedts_data_source column to each table to record data sources. In this case, DDL operations cannot be synchronized. No is selected by default. In this case, DDL operations can be synchronized. Note You can merge the data source columns based on tasks rather than tables. To merge only the data source columns of some tables, you can create two data synchronization tasks. | |
| Synchronization Type | Select the types of operations that you want to synchronize based on your business requirements. All operation types are selected by default. Note Only INSERT, UPDATE, DELETE, and ADD COLUMN operations can be synchronized. | |



- 9. In the lower-right corner of the page, click Next.
- 10. Specify a type for the tables to be synchronized to the destination database.



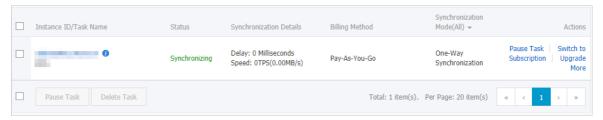
Note After you select Initial Schema Synchronization, you must specify the type, primary key column, and partition key column for the tables to be synchronized to AnalyticDB for MySQL. For more information, see CREATE TABLE.

11. In the lower-right corner of the page, click Precheck.

- ? Note
 - Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
 - If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

You can view the status of the data synchronization task on the Data Synchronization page.



Troubleshoot the synchronization failure that occurs due to field type changes

In this example, the data of a table named customer fails to be synchronized to the destination AnalyticDB for MySQL cluster.

- 1. In the destination AnalyticDB for MySQL cluster, create a table named customer_new with the same schema as the customer table.
- 2. Run the INSERT INTO SELECT command to copy the data of the customer table and insert the data into the customer_new table. This ensures that the data of the two tables is consistent.
- 3. Rename or delete the customer table. Then, change the name of the customer_new table to customer.
- 4. Restart the data synchronization task in the DTS console.

7.3. Synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for PostgreSQL instance

This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for PostgreSQL instance by using Data Transmission Service (DTS). The data synchronization feature provided by DTS allows you to transfer and analyze data with ease.

Prerequisites

- The tables to be synchronized from the ApsaraDB RDS for MySQL instance contain primary keys.
- An AnalyticDB for PostgreSQL instance is created. For more information, see Create an instance.

Notes

DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU loads of the source and destination databases are less than 30%.

Limits

- You can select only tables as the objects to be synchronized.
- You cannot synchronize the following types of data: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID SNAPSHOT.
- We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on objects during data synchronization. Otherwise, data synchronization may fail.

Supported SQL operations

- DML operations: INSERT, UPDATE, and DELETE
- DDL operations: ADD COLUMN, and RENAME COLUMN

Note The CREATE TABLE operation is not supported. To synchronize data from a new table, you must add the table to the selected objects. For more information, see Add an object to a data synchronization task.

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way many-to-one synchronization

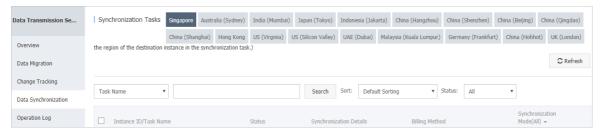
Term mappings

| Term in MySQL | Term in AnalyticDB for PostgreSQL |
|---------------|-----------------------------------|
| Database | Schema |
| Table | Table |

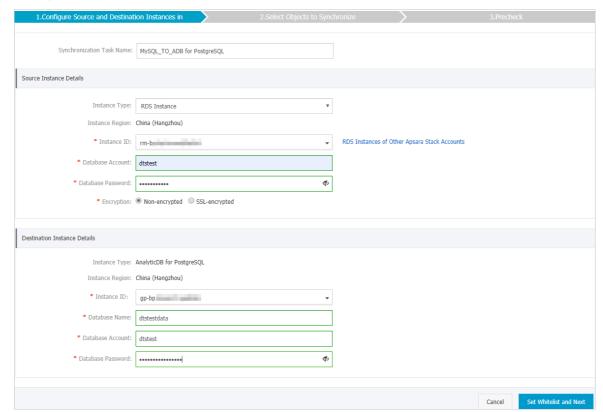
Procedure

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the purchase page, select MySQL for the source instance and AnalyticDB for PostgreSQL for the destination instance. Select One-Way Synchronization as the synchronization topology.
- 2. Log on to the DTS console.

- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



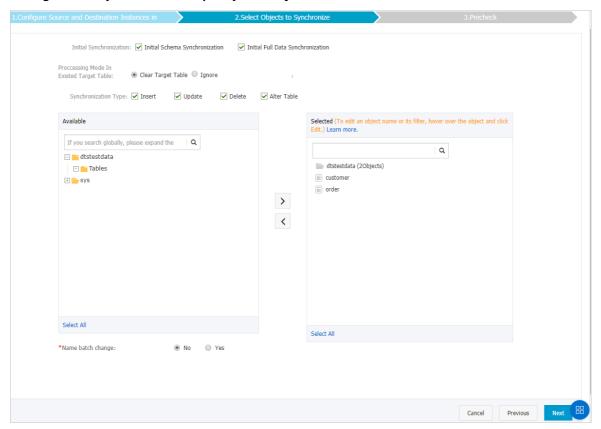
| Section | Parameter | Description |
|---------|----------------------------------|---|
| N/A | Synchronizat ion Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select RDS Instance. |
| | Instance Region | The region of the source instance. The value is the same as that you selected when purchasing the data synchronization instance. You cannot change the value of this parameter. |
| | | |

| Section | Parameter | Description | |
|------------------------------------|----------------------|---|--|
| | Instance ID | Select the ID of the source RDS instance. | |
| Source Instance Details | Database Account | Enter the database account for the source ApsaraDB RDS for MySQL instance. Note If the database type of the source RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account and database password. | |
| | Database Password | Enter the password for the database account. | |
| | Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. | |
| | | Note The Encryption parameter is available only in mainland China and Hong Kong(China). | |
| | Instance Type | The value of this parameter is set to AnalyticDB for PostgreSQL and cannot be changed. | |
| | Instance Region | The region of the destination instance. The value is the same as that you selected when purchasing the data synchronization instance. You cannot change the value of this parameter. | |
| | Instance ID | Select the ID of the destination AnalyticDB for PostgreSQL instance. | |
| Destination Instance Details | Database Name | The name of the destination database. | |
| DETAILS | Database Account | Enter the database account for the destination AnalyticDB for PostgreSQL instance. Note The database account must have the SELECT, INSERT, UPDATE, DELETE, COPY, TRUNCATE, and ALTER TABLE permissions. | |
| | Database Password | Enter the password for the database account. | |

7. In the lower-right corner of the page, click **Set Whitelist and Next**.

Note The CIDR blocks of DTS servers are automatically added to the whitelists of the source and destination instances. This ensures that DTS servers can connect to the source and destination instances.

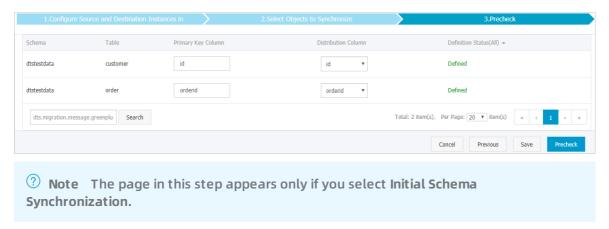
8. Configure the synchronization policy and objects.



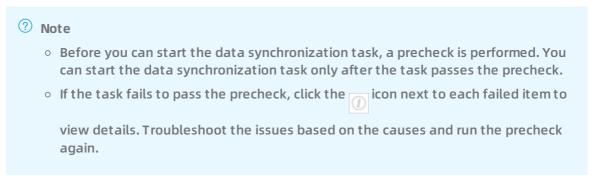
| Section | Parameter | Description |
|---------|-------------------------|---|
| | Initial Synchronization | You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization. |
| | | |
| | | |
| | | |
| | | |

| Section | Parameter | Description |
|--|---|---|
| Synchronizat ion policy Processing Mode In Existed Target Table Synchronization Type | Clear Target Table Skips the Schema Name Conflict item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. Ignore Skips the Schema Name Conflict item during the precheck. Adds new data to the existing data during initial full data synchronization. You can select this mode if you want to synchronize data from multiple tables to one table. | |
| | Synchronization Type | Select the types of operations that you want to synchronize based on your business requirements. Insert Update Delete Alter Table |
| Objects to be synchronized | N/A | Select tables from the Available section and click the right arrow () icon to add the tables to the Selected section. Note You can select only tables as the objects to be synchronized. You can change the names of columns in the destination database by using the object name mapping feature provided by DTS. For more information about how to use this feature, see Specify the name of an object in the destination instance. |

9. Specify the primary key column and distribution column of the table that you want to synchronize to the AnalyticDB for PostgreSQL instance.

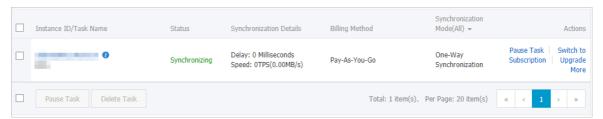


10. In the lower-right corner of the page, click Precheck.



- 11. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 12. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the Synchronization Tasks page, view the status of the data synchronization task.



7.4. Synchronize data from a user-created MySQL database hosted on ECS to an AnalyticDB for PostgreSQL instance

This topic describes how to synchronize data from a user-created MySQL database hosted on ECS to an AnalyticDB for PostgreSQL instance by using Data Transmission Service (DTS).

Prerequisites

- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The binary logging feature is enabled for the source database. A database account is created

for the data synchronization task. For more information, see Create an account for a user-created MySQL database and configure binary logging.

Note The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.

- The tables to be synchronized from the source database contain primary keys.
- An AnalyticDB for PostgreSQL instance is created. For more information, see Create an instance.

Notes

DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU loads of the source and destination databases are less than 30%.

Limits

- You can select only tables as the objects to be synchronized.
- You cannot synchronize the following types of data: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID_SNAPSHOT.
- We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on objects during data synchronization. Otherwise, data synchronization may fail.

Supported SQL operations

- DML operations: INSERT, UPDATE, and DELETE
- DDL operations: ADD COLUMN, and RENAME COLUMN

Note The CREATE TABLE operation is not supported. To synchronize data from a new table, you must add the table to the selected objects. For more information, see Add an object to a data synchronization task.

Supported synchronization topologies

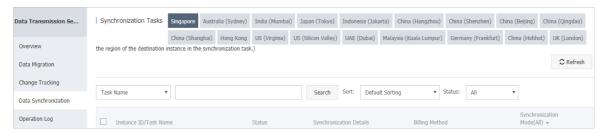
- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way many-to-one synchronization

Term mappings

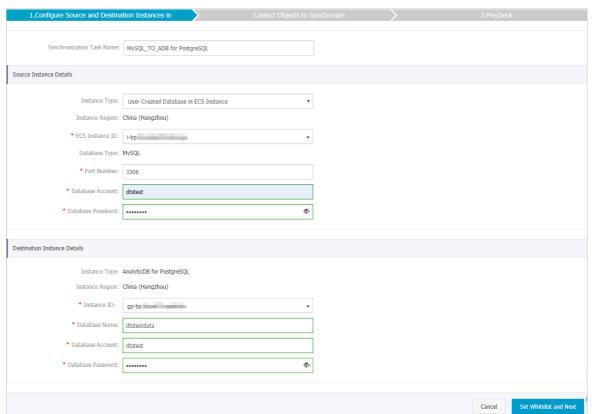
| Term in MySQL | Term in AnalyticDB for PostgreSQL |
|---------------|-----------------------------------|
| Database | Schema |
| Table | Table |

Procedure

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the purchase page, select MySQL for the source instance and AnalyticDB for PostgreSQL for the destination instance. Select One-Way Synchronization as the synchronization topology.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



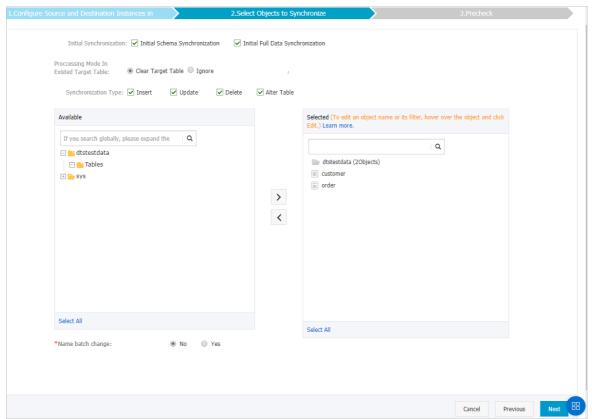
| Section | Parameter | Description | |
|-------------------------------|-------------------------------------|---|--|
| N/A | Synchroni zation Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. | |
| | Instance Type | Select User-Created Database in ECS Instance. | |
| | Instance Region | The region of the source instance. The value is the same as that you selected when purchasing the data synchronization instance. You cannot change the value of this parameter. | |
| | Instance ID | Select the ID of the ECS instance that is connected to the user-created MySQL database. | |
| | Database Type | The value of this parameter is set to MySQL and cannot be changed. | |
| | Port Number | Enter the service port number of the user-created MySQL database. The default port number is 3306. | |
| Source Instance Details | | | |

| Section | Parameter | Description | |
|-------------------------|----------------------|--|--|
| | | Enter the account for the user-created MySQL database. | |
| 2414.5 | Database Account | Note The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. | |
| | Database Password | Enter the password for the database account. | |
| | Instance Type | The value of this parameter is set to AnalyticDB for PostgreSQL and cannot be changed. | |
| | Instance Region | The region of the destination instance. The value is the same as that you selected when purchasing the data synchronization instance. You cannot change the value of this parameter. | |
| | Instance ID | Select the ID of the destination AnalyticDB for PostgreSQL instance. | |
| Destination Instance | Database Name | Enter the name of the destination database. | |
| Details | | Enter the database account for the destination AnalyticDB for PostgreSQL instance. | |
| | Database Account | Note The database account must have the SELECT, INSERT, UPDATE, DELETE, COPY, TRUNCATE, and ALTER TABLE permissions. | |
| | Database Password | Enter the password for the database account. | |

7. In the lower-right corner of the page, click Set Whitelist and Next.

Note The CIDR blocks of DTS servers are automatically added to the inbound rule of the source ECS instance and the whitelist of the destination AnalyticDB for PostgreSQL instance. This ensures that DTS servers can connect to the source and destination instances.

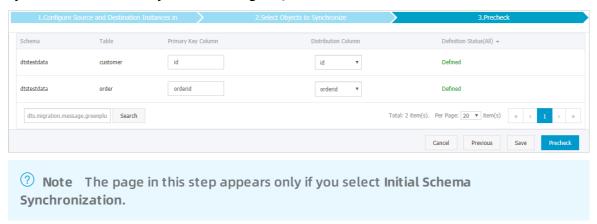
8. Configure the synchronization policy and objects.



| Section | Parameter | Description |
|-------------------------|---|--|
| Initial Synchronization | You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization. | |
| | Processing Mode In Existed Target Table | Clear Target Table Skips the Schema Name Conflict item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. Ignore |
| | | Skips the Schema Name Conflict item during the precheck. Adds new data to the existing data during initial full data synchronization. You can select this mode if you want to synchronize data from multiple tables to one table. |

| Section | Parameter | Description |
|--------------------------------|---|---|
| | Synchronization Type | Select the types of operations that you want to synchronize based on your business requirements. Insert Update Delete Alter Table |
| Objects to be synchronized N/A | | Select tables from the Available section and click the right arrow (>) icon to add the tables to the Selected section. |
| | Note You can select only tables as the objects to be synchronized. You can change the names of columns in the destination database by using the object name mapping feature provided by DTS. For more information about how to use this feature, see Specify the name of an object in the destination instance. | |

9. Specify the primary key column and distribution column of the table that you want to synchronize to the AnalyticDB for PostgreSQL instance.

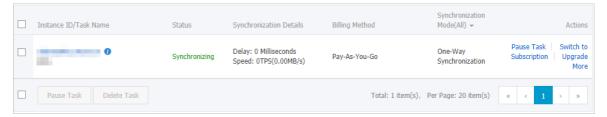


10. In the lower-right corner of the page, click **Precheck**.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 11. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 12. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.

On the Synchronization Tasks page, view the status of the data synchronization task.



7.5. Synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an AnalyticDB for PostgreSQL instance

This topic describes how to synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an AnalyticDB for PostgreSQL instance by using Data Transmission Service (DTS). The data synchronization feature allows you to transfer and analyze data with ease.

Prerequisites

- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The tables to be synchronized from the source database contain primary keys.
- The binary logging feature is enabled for the source database. A database account is created for the data synchronization task. For more information, see Create an account for a user-created MySQL database and configure binary logging.

② Note The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.

- The on-premises network to which the user-created MySQL database belongs is connected to Alibaba Cloud VPC over Express Connect, VPN Gateway, or Smart Access Gateway. DTS is allowed to access the network to which Express Connect, VPN Gateway, or Smart Access Gateway belongs. For more information, see Configure a route between DTS and Express Connect, VPN Gateway, or Smart Access Gateway.
 - Note For more information about how to connect a VPC to an on-premises data center, see Connect to local IDCs.
- An AnalyticDB for PostgreSQL instance is created. For more information, see Create an
 instance.

Precautions

DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before you synchronize data, evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.

Limits

- You can select only tables as the objects to be synchronized.
- You cannot synchronize the following types of data: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID_SNAPSHOT.
- We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on objects during data synchronization. Otherwise, data synchronization may fail.

SQL operations that can be synchronized

- DML operations: INSERT, UPDATE, and DELETE
- DDL operations: ADD COLUMN and RENAME COLUMN

Note The CREATE TABLE operation is not supported. To synchronize data from a new table, you must add the table to the selected objects. For more information, see Add an object to a data synchronization task.

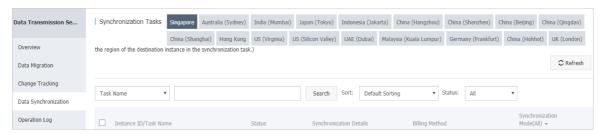
Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way many-to-one synchronization

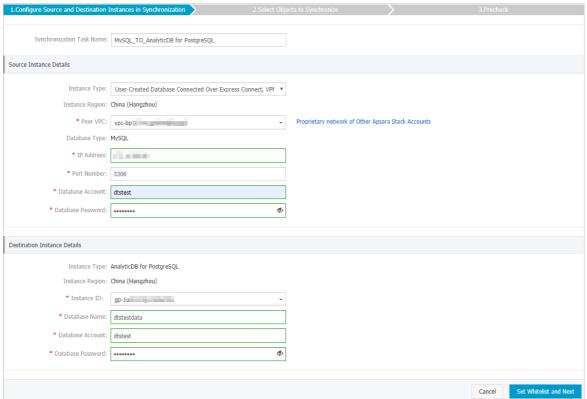
Term mappings

| Term in MySQL | Term in AnalyticDB for PostgreSQL | |
|---------------|-----------------------------------|--|
| Database | Schema | |
| Table | Table | |

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to MySQL, set Target Instance to AnalyticDB for PostgreSQL, and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



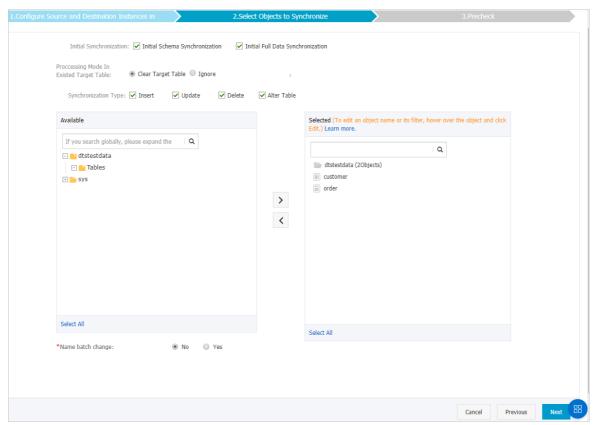
| Section | Parameter | Description | |
|---------------------|----------------------------------|---|--|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. | |
| | Instance Type | Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway. | |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. | |
| | Peer VPC | Select the ID of the VPC that is connected to the user-created MySQL database. | |
| Source | Database Type | The value of this parameter is set to MySQL and cannot be changed. | |
| | IP Address | Enter the server IP address of the user-created MySQL database. | |
| | Port Number | Enter the service port number of the user-created MySQL database. The default port number is 3306. | |
| Instance Details | | | |

| Section | Parameter | Description | |
|------------------------------------|----------------------|---|--|
| | | Enter the account of the user-created MySQL database. | |
| | Database Account | Note The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. | |
| | Database Password | Enter the password for the source database account. | |
| | Instance Type | The value of this parameter is set to AnalyticDB for PostgreSQL and cannot be changed. | |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. | |
| | Instance ID | Select the ID of the destination AnalyticDB for PostgreSQL instance. | |
| Destination Instance Details | Database Name | Enter the name of the destination database. | |
| | Database Account | Enter the database account of the destination AnalyticDB for PostgreSQL instance. | |
| | | Note The account must have the SELECT, INSERT, UPDATE, DELETE, COPY, TRUNCATE, and ALTER TABLE permissions. | |
| | Database Password | Enter the password for the destination database account. | |

7. In the lower-right corner of the page, click **Set Whitelist and Next**.

Note The CIDR blocks of DTS servers are automatically added to the whitelist of the destination AnalyticDB for PostgreSQL instance. This ensures that DTS servers can connect to the destination AnalyticDB for PostgreSQL instance.

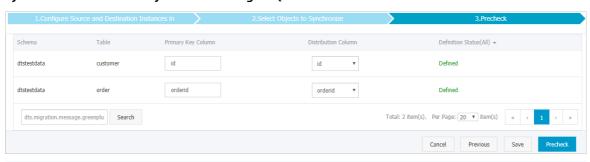
8. Configure the synchronization policy and objects.



| Section | Parameter | Description |
|-------------------------|--|--|
| | Initial Synchronization | You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization. |
| | Processing Mode In Existed Target Table | Clear Target Table Skips the Schema Name Conflict item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. Ignore |
| Synchronizat ion policy | | Skips the Schema Name Conflict item during the precheck. Adds data to the existing data during initial full data synchronization. If you want to synchronize data from multiple tables to one table, you can select this mode. |

| Section | Parameter | Description | | |
|-------------------------------|----------------------|---|--|--|
| | Synchronization Type | Select the types of operations that you want to synchronize based on your business requirements. Insert Update Delete Alter Table | | |
| Objects to be synchronized | N/A | Alter Table Select tables from the Available section and click th icon to move the tables to the Selected section. Note You can select only tables as the objects to be synchronized. You can change the names of the columns that are synchronized to the destination database by using the object name mapping feature. For more information about how to use this feature, see Specify the name of an object in the destination instance. | | |

9. Specify the primary key column and distribution column of the table that you want to synchronize to the AnalyticDB for PostgreSQL instance.



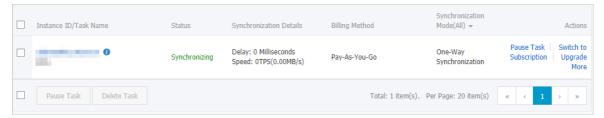
Note The page in this step appears only if you select Initial Schema Synchronization. For more information about primary key columns and distribution columns, see Table constraints and Partition keys.

10. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 11. Close the Precheck dialog box after the following message is displayed: The precheck is passed. Then, the data synchronization task starts.
- 12. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.

You can view the status of the data synchronization task on the **Synchronization Tasks** page.



7.6. Synchronize data from a user-created MySQL database hosted on ECS to Elasticsearch

Alibaba Cloud Elasticsearch is compatible with open-source Elasticsearch features such as Security, Machine Learning, Graph, and Application Performance Management (APM). Alibaba Cloud Elasticsearch provides capabilities such as enterprise-level access control, security monitoring and alerts, and automatic report generation. You can use Alibaba Cloud Elasticsearch to search and analyze data. You can use Data Transmission Service (DTS) to synchronize data from a user-created MySQL database that is hosted on ECS to an Elasticsearch instance.

Prerequisites

- An Elasticsearch instance (5.5, 6.3, or 6.7 version) is created. For more information, see Create an Alibaba Cloud Elasticsearch instance.
- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.

Limits

DTS uses read and write resources of the source and destination databases during initial full
data synchronization. This may increase the database load. If the database performance is
unfavorable, the specification is low, or the data volume is large, database services may
become unavailable. For example, DTS occupies a large amount of read and write resources in
the following cases: a large number of slow SQL queries are performed on the source

database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.

- DDL operations cannot be synchronized. If a DDL operation is performed on the table in the source database during data synchronization, you must perform the following steps: Remove the table from the required objects, remove the index for the table from the Elasticsearch instance, and then add the table to the required objects. For more information, see Remove an object from a data synchronization task and Add an object to a data synchronization task.
- To add columns to the table that you want to synchronize, perform the following steps: Modify the mappings of the table in the Elasticsearch instance, perform DDL operations in the source MySQL database, and then pause and start the data synchronization task.

SQL operations supported by data synchronization

INSERT, DELETE, and UPDATE operations

Data type mappings

The data types of MySQL and Elasticsearch instances do not have one-to-one correspondence. During initial schema synchronization, DTS maps the data types of the source database to the destination database. For more information, see Data type mappings for initial schema synchronization.

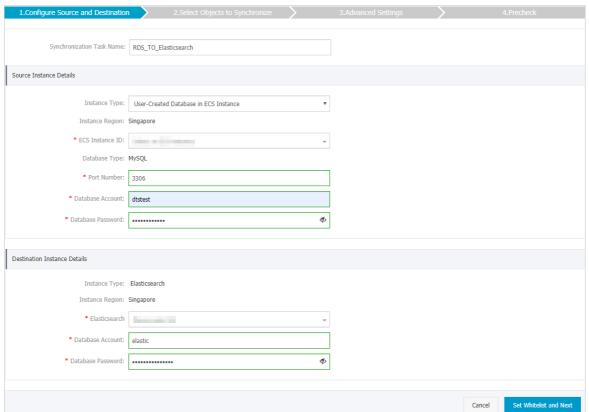
Preparations

Create an account for a user-created MySQL database and configure binary logging

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note Select MySQL for the source instance and Elasticsearch for the destination instance. Select One-Way Synchronization as the synchronization topology.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



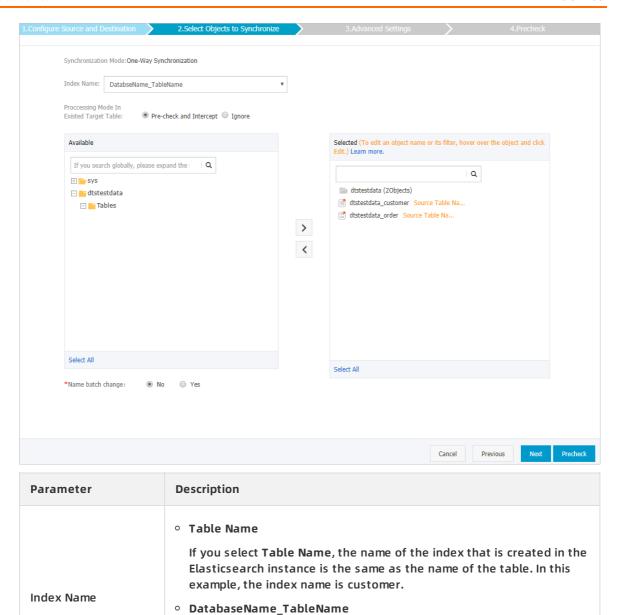
- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



| Section | Parameter | Description | |
|-------------------------------|----------------------------------|--|--|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. | |
| | Instance Type | Select User-Created Database in ECS Instance. | |
| Instance Region | | The region of the source instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. | |
| | ECS Instance ID | Select the ID of the ECS instance that is connected to the user-created MySQL database. | |
| | Database Type | The value of this parameter is set to MySQL and cannot be changed. | |
| | Port Number | Enter the service port number of the user-created MySQL database. | |
| Source Instance Details | | | |

| Section | Parameter | Description | |
|------------------------------------|----------------------|---|--|
| | | Enter the account for the user-created MySQL database. | |
| | Database Account | Note The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. | |
| | Database Password | Enter the password for the database account. | |
| Destination Instance Details | Instance Type | The value of this parameter is set to Elasticsearch and cannot be changed. | |
| | Instance Region | The region of the destination instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. | |
| | Elasticsearc h | Select the ID of the Elasticsearch instance. | |
| | Database Account | Enter the account that is used to connect to the Elasticsearch instance. The default account is elastic. | |
| | Database Password | Enter the password for the account. | |

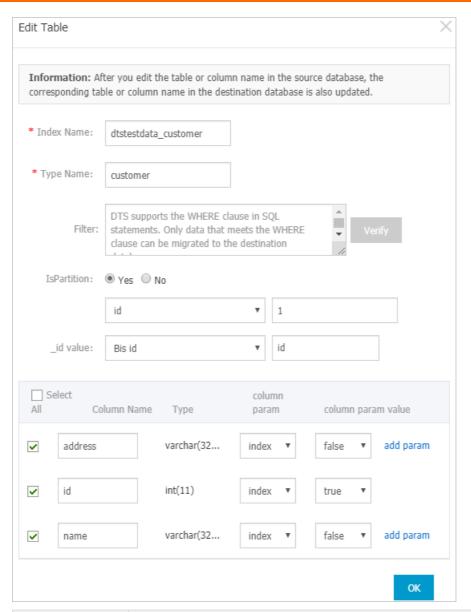
- 7. In the lower-right corner of the page, click Set Whitelist and Next.
 - Note The CIDR blocks of DTS servers are automatically added to the inbound rule of the ECS instance and the whitelist of the Elasticsearch instance. This ensures that DTS servers can connect to the source and destination instances.
- 8. Configure the index name, the processing mode of identical index names, and the objects to be synchronized.



If you select <code>DatabaseName_TableName</code>, the name of the index that is created in the Elasticsearch instance is <code><Database</code> name<code>>_<Table</code> name<code>></code>. In this example, the index name is <code>dtstestdata_customer</code>.

| Parameter | Description | | |
|--|--|--|--|
| | Pre-check and Intercept: checks whether the destination database contains indexes that have the same names as tables in the source database. If the destination database does not contain indexes that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. Note If indexes in the destination database have the same names as tables in the source database, and cannot be deleted | | |
| | or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. | | |
| | Ignore: skips the precheck for identical index names in the source and destination databases. | | |
| Processing Mode In Existed Target Table | Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. If the source and destination databases have the same mappings and the primary key of a record in the destination database is the same as that in the source database, the record remains unchanged during initial data synchronization. However, the record is overwritten during incremental data synchronization. If the source and destination databases have different | | |
| | mappings, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. | | |
| | | | |
| Objects to be synchronized | Select objects from the Available section and click the icon to move the objects to the Selected section. You can select tables and databases as the objects to be synchronized. | | |

9. In the Selected section, move the pointer over a table, and then click Edit. In the Edit Table dialog box that appears, configure parameters for the table in the Elasticsearch instance, such as the index name and type name.



| Parameter | Description | | |
|-------------|--|--|--|
| Index Name | For more information, see Terms in the Elasticsearch documentation. | | |
| Type Name | Warning The only type of special characters that an index name and type name can contain is underscore (_). An index name and type name cannot contain the following special characters: +-&!(){} ^"~*?:\% . | | |
| Filter | Specify SQL conditions to filter data. Only data that meets the specified conditions is synchronized to the destination instance. For more information, see Use SQL conditions to filter data. | | |
| IsPartition | Select whether to set partitions. If you select Yes, you must also specify the partition key column and number of partitions. | | |

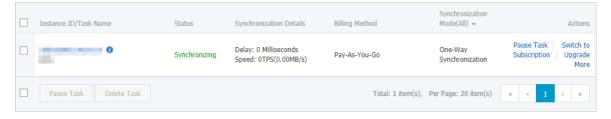
| Parameter | Description | |
|-----------|--|--|
| _id value | Primary key column Composite primary key fields are merged into one column. Business key If you select a business key, you must also specify the business key column. | |
| add param | Select the column parameter and parameter value. For more information, see Mapping parameters in the Elasticsearch documentation. | |

10. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 11. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 12. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

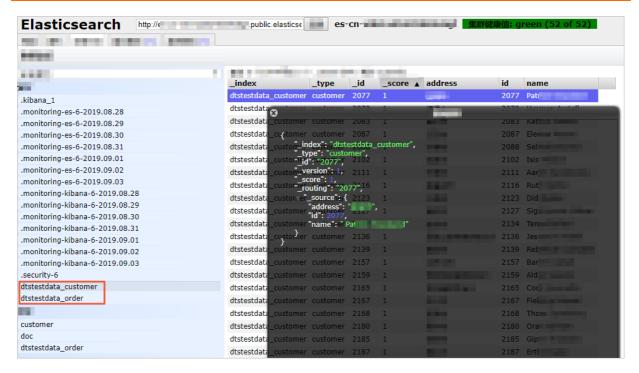
On the Synchronization Tasks page, view the status of the data synchronization task.



Check the index and data

If the data synchronization task is in the **Synchronizing** state, you can connect to the Elasticsearch instance and check whether the index is created and data is synchronized as expected. In this example, the **Elasticsearch Head** plug-in is used.

? Note If the index is not created or data is not synchronized as expected, you can delete the index and data, and then configure the data synchronization task again.



7.7. Synchronize data from an ApsaraDB RDS MySQL instance to a MaxCompute project

MaxCompute (previously known as ODPS) is a fast and fully managed computing platform for large-scale data warehousing. MaxCompute can process exabytes of data. This topic describes how to synchronize data from an ApsaraDB RDS MySQL instance to a MaxCompute project by using Data Transmission Service (DTS).

Prerequisites

- MaxCompute is activated. For more information, see Activate MaxCompute.
- A project is created in MaxCompute. For more information, see Create a project.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- Only table-level data can be synchronized.
- We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL

operations on objects during data synchronization. Otherwise, data synchronization may fail.

• MaxCompute does not support the PRIMARY KEY constraint. If network errors occur, DTS may synchronize duplicate data records to MaxCompute.

Supported source database types

You can use DTS to synchronize data from the following types of MySQL databases:

- User-created database hosted on ECS
- User-created database connected over Express Connect, VPN Gateway, or Smart Access Gateway
- User-created database connected over a database gateway
- ApsaraDB RDS MySQL instance that is owned by the same Alibaba Cloud account as MaxCompute or a different Alibaba Cloud account from MaxCompute

This topic uses an **ApsaraDB RDS MySQL instance** as an example to describe how to configure a data synchronization task. You can also follow the procedure to configure data synchronization tasks for other types of MySQL databases.

Note If your source database is a user-created MySQL database, you must prepare the environments that are required for the source database. For more information, see Preparation overview.

SQL operations that can be synchronized

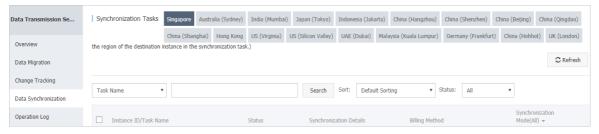
• DDL operation: ADD COLUMN

• DML operations: INSERT, UPDATE, and DELETE

Synchronization process

- Initial schema synchronization DTS synchronizes the schemas of the required objects from
 the source database to MaxCompute. During initial schema synchronization, DTS adds the
 _base suffix to the end of the source table name. For example, if the name of the source
 table is customer, the name of the table in MaxCompute is customer_base.
- 2. Initial full data synchronization DTS synchronizes the historical data of the table from the source database to the destination table in MaxCompute. For example, the customer table in the source database is synchronized to the customer_base table in MaxCompute. The data is the basis for subsequent incremental synchronization.
 - ? Note The destination table that is suffixed with _base is known as a full baseline table.
- 3. Incremental data synchronization DTS creates an incremental data table in MaxCompute. The name of the incremental data table is suffixed with _log, such as customer_log. Then, DTS synchronizes incremental data that is generated in the source database to the incremental data table in real time.
 - Note For more information, see Schema of an incremental data table.

- Warning To ensure that the synchronization account is authorized, we recommend that you perform the following steps by using your Alibaba Cloud account.
- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to MySQL, set Target Instance to MaxCompute, and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.
- 7. In the lower-right corner of the page, click Set Whitelist and Next.
 - ? Note The CIDR blocks of DTS servers are automatically added to the whitelist of the source RDS instance and the MaxCompute project. This ensures that DTS servers can connect to the source and destination instances.
- 8. In the lower-right corner of the page, click **Next**. In this step, the permissions on the MaxCompute project are granted to the synchronization account.

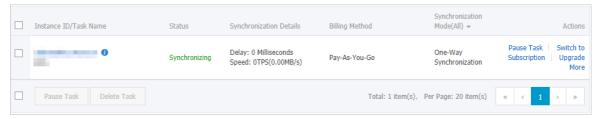


- 9. Configure the synchronization policy and objects.
- 10. In the lower-right corner of the page, click Precheck.



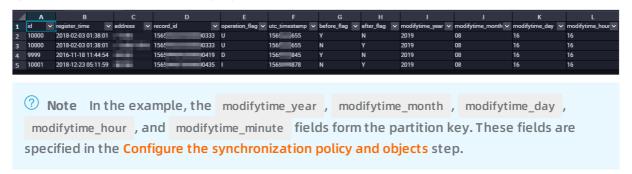
- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 11. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 12. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.

On the Synchronization Tasks page, view the status of the data synchronization task.

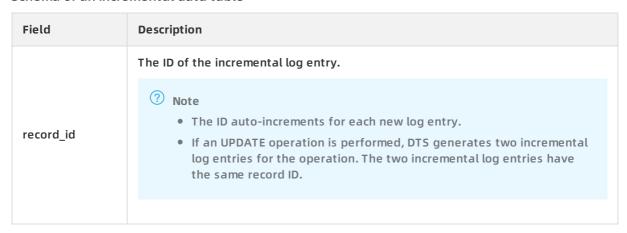


Schema of an incremental data table

DTS synchronizes incremental data that is generated in the source MySQL database to the incremental data table in MaxCompute. The incremental data table stores incremental data and specific metadata. The following figure shows the schema of an incremental data table.



Schema of an incremental data table



| Field | Description | | |
|----------------|---|--|--|
| operation_flag | The operation type. Valid values: I: an INSERT operation. D: a DELETE operation. U: an UPDATE operation. | | |
| utc_timestamp | The operation timestamp. It is also the timestamp of the binary log file. The timestamp is in the UTC format. | | |
| before_flag | Indicates whether the column values are pre-update values. Valid values: Y and N. | | |
| after_flag | Indicates whether the column values are post-update values. Valid values: Y and N. | | |

Additional information about the before_flag and after_flag fields

For different operation types, the before_flag and after_flag fields of an incremental log entry are defined as follows:

INSERT

For an INSERT operation, the column values are the newly inserted record values (post-update values). The value of the before_flag field is N and the value of the after_flag field is Y.



UPDATE

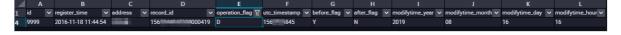
DTS generates two incremental log entries for an UPDATE operation. The two incremental log entries have the same values for the record_id, operation_flag, and dts_utc_timestamp fields.

The second log entry records the pre-update values, so the value of the before_flag field is Y and the value of the after_flag field is N. The second log entry records the post-update values, so the value of the before_flag field is N and the value of the after_flag field is Y.



DELETE

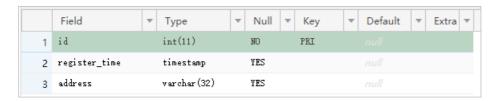
For a DELETE operation, the column values are the deleted record values (pre-update values). The value of the before_flag field is Y and the value of the after_flag field is N.



Merge a full baseline table and incremental data table

After a data synchronization task is started, DTS creates a full baseline table and an incremental data table in MaxCompute. You can use SQL statements to merge the two tables. This allows you to obtain the full data at a specific time point.

This section describes how to merge data for the customer table. The following figure shows the schema of the customer table.



Procedure

1. Create a table in MaxCompute based on the schema of the source table. The table is used to store the merged data.

For example, you can obtain full data of the customer table at the 1565944878 time point. Run the following SQL statements to create the required table:

```
CREATE TABLE `customer_1565944878` (
  `id` bigint NULL,
  `register_time` datetime NULL,
  `address` string);
```

? Note

- You can use the ad-hoc query feature to run SQL statements. For more information, see (Optional) Use an ad-hoc query to run SQL statements.
- For more information about the data types that are supported by MaxCompute, see Data types.
- 2. Run the following SQL statements in MaxCompute to merge the full baseline table and incremental data table and obtain full data at a specific time point:

```
set odps.sql.allow.fullscan=true;
insert overwrite table <result_storage_table>
select <col1>,
<col2>,
<colN>
from(
select row_number() over(partition by t.<primary_key_column>
order by record_id desc, after_flag desc) as row_number, record_id, operation_flag, after_flag, <co
l1>, <col2>, <colN>
from(
select incr.record_id, incr.operation_flag, incr.after_flag, incr.<col1>, incr.<col2>,incr.<colN>
from <table_log> incr
where utc_timestamp< <timestamp>
select 0 as record_id, 'I' as operation_flag, 'Y' as after_flag, base.<col1>, base.<col2>,base.<colN>
from <table_base> base) t) gt
where record_num=1
and after flag='Y'
```

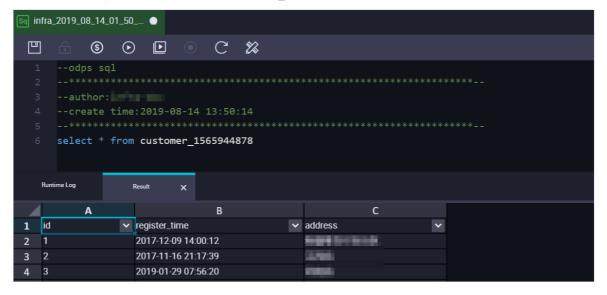
? Note

- <result_storage_table>: the name of the table that stores the merged data.
- <col1>/<col2>/<colN>: the names of the columns in the table to be merged.
- <primary_key_column>: the name of the primary key column in the table to be merged.
- <table_log>: the name of the incremental data table.
- <table_base>: the name of the full baseline table.
- <timestamp>: the timestamp that is generated when full data is obtained.

Run the following SQL statements to obtain full data of the customer table at the 156594487 time point:

```
set odps.sql.allow.fullscan=true;
insert overwrite table customer_1565944878
select id,
register_time,
address
from(
select row_number() over(partition by t.id
order by record_id desc, after_flag desc) as row_number, record_id, operation_flag, after_flag, id,
register_time, address
select incr.record_id, incr.operation_flag, incr.after_flag, incr.id, incr.register_time, incr.address
from customer_log incr
where utc_timestamp< 1565944878
select 0 as record_id, 'I' as operation_flag, 'Y' as after_flag, base.id, base.register_time, base.addr
from customer_base base) t) gt
where gt.row number= 1
and gt.after_flag= 'Y';
```

3. Query the merged data from the customer_1565944878 table.



7.8. Use alternative naming for systemcreated columns

Data Transmission Service (DTS) creates metadata columns for the tables that are replicated to MaxCompute. These additional columns are used to store metadata that assists the data processing of MaxCompute, such as sorting and removal of duplicates. If your source table contains columns that have the same names as the metadata columns created by DTS, the data synchronization task fails. To resolve this issue, you can configure DTS to use alternative naming during the creation of metadata columns.

Context

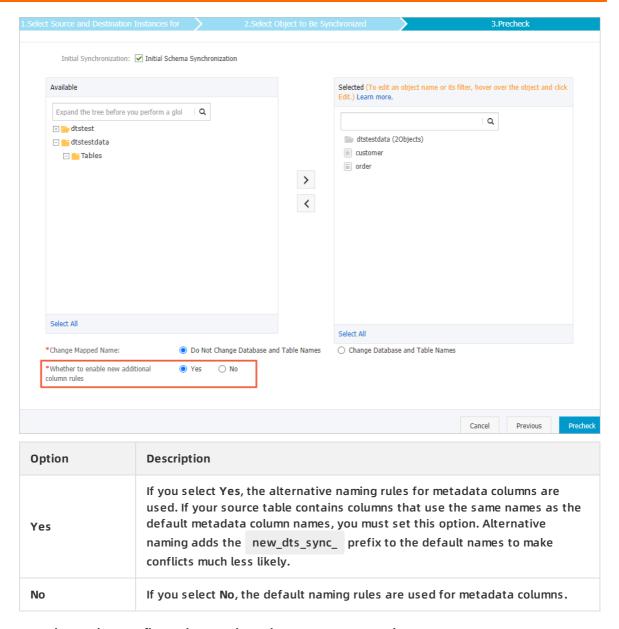
Before you enable alternative naming for metadata columns, check whether the existing columns in your source table have the same names as the default column names used by DTS as listed in the following table.

Note In step 2, you can specify whether to use the alternative naming rules for metadata columns.

Naming rules for metadata columns

| Target instance | Default column names | Alternative column names | Reference |
|--------------------|--|--|--|
| MaxComput e | record_id operation_flag utc_timestamp before_flag after_flag modifytime_year modifytime_mont h modifytime_day modifytime_hour modifytime_minut e | new_dts_sync_record_id new_dts_sync_operation_flag new_dts_sync_utc_timestamp new_dts_sync_before_flag new_dts_sync_after_flag new_dts_sync_modifytime_year new_dts_sync_modifytime_month new_dts_sync_modifytime_day new_dts_sync_modifytime_hour new_dts_sync_modifytime_minute | For more information, see Schema of an incremental data table. |

- 1. Create and configure a data synchronization task. For more information, see steps 1 to 7 in Create a data synchronization task.
- 2. In the Select Objects to Synchronize step, specify whether you want to use the new naming rules for additional columns.



3. Complete other configurations and continue to create a task.

References

- Database types, initial synchronization types, and synchronization topologies
- Overview of data synchronization scenarios

7.9. Synchronize data from a user-created MySQL database to a Message Queue for Apache Kafka instance

This topic describes how to synchronize data from a user-created MySQL database to a Message Queue for Apache Kafka instance by using Data Transmission Service (DTS). This allows you to extend message processing capabilities.

Prerequisites

- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The version of the destination Kafka instance is 0.10 or 1.0.
- In the destination Kafka instance, a topic is created to receive the synchronized data. For more information, see Create a topic.

Context

Message Queue for Apache Kafka is a distributed, high-throughput, and scalable message queue service provided by Alibaba Cloud. It provides fully managed services for the open-source Apache Kafka to solve the long-standing shortcomings of open-source products. Message Queue for Apache Kafka allows you to focus on business development without the need to consider deployment and O&M. Message Queue for Apache Kafka is widely used in big data fields such as log collection, monitoring data aggregation, streaming data processing, and online and offline analysis. It has become indispensable for the big data ecosystem.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination database.

Limits

- You can select only tables as the objects to be synchronized.
- DTS does not synchronize the data in a renamed table to the destination Kafka cluster. This applies if the new table name is not included in the objects to be synchronized. To synchronize the data in a renamed table to the destination Kafka cluster, you must modify the objects to be synchronized. For more information, see Add an object to a data synchronization task.

Data format

The data that is synchronized to the Kafka cluster is stored in the Avro format. For more information, see DTS Avro schema.

After the data is synchronized to the Kafka cluster, you must parse the data based on the Avro schema.

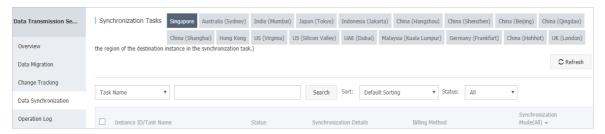
Billing

For more information, see Pricing.

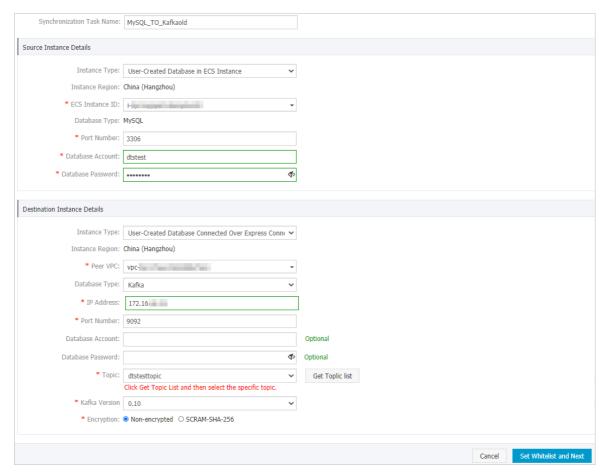
Preparations

Create an account for a user-created MySQL database and configure binary logging

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to MySQL, set Target Instance to Kafka, and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

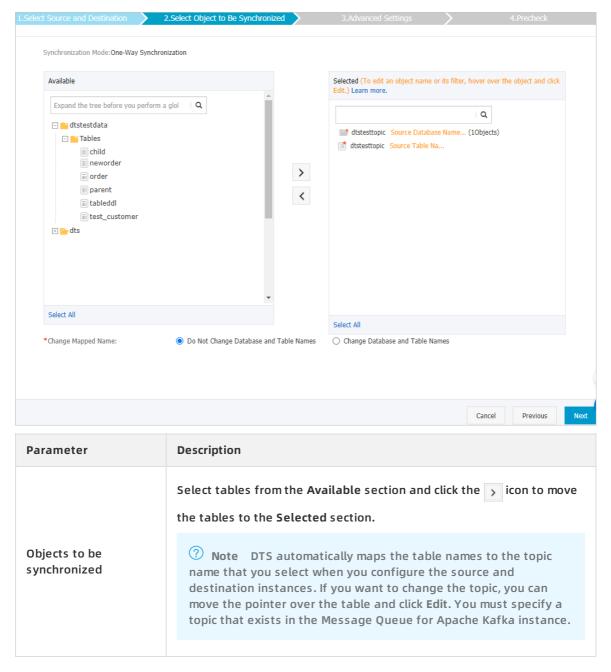


| Section | Parameter | Description |
|-------------------------------|----------------------------------|--|
| N/A | Synchroniza tion Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| Source Instance Details | Instance Type | Select RDS Instance, User-Created Database in ECS Instance, or User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway based on where the source database is deployed. |
| | | This topic uses User-Created Database in ECS Instance as an example to describe how to configure a data synchronization task. You can also follow the procedure to configure data synchronization tasks for other types of user-created MySQL databases. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | ECS Instance ID | Select the ID of the ECS instance that is connected to the user-created MySQL database. |
| | | |

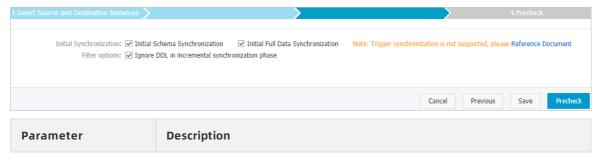
| Section | Parameter | Description |
|---------|----------------------|--|
| | Database Type | The value of this parameter is set to MySQL and cannot be changed. |
| | Port Number | Enter the service port number of the user-created MySQL database. |
| | Database Account | Enter the account of the user-created MySQL database. The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. |
| | Database Password | Enter the password for the database account. |
| | | Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway. |
| | Instance Type | Note You cannot select Message Queue for Apache Kafka as the instance type. You can use Message Queue for Apache Kafka as a user-created Kafka database to configure data synchronization. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | | Select the ID of the VPC to which the destination Message Queue for Apache Kafka instance belongs. To obtain the VPC ID, you can log on to the Message Queue for Apache Kafka console and navigate to the Instance Details page of the Message Queue for Apache Kafka instance. In the Basic Information section, you can view the VPC ID. |
| | Peer VPC | Instance Details Task Records Basic Information Instance ID: Instance Type: Standard Edition Peak Traffic: 20 MB/s Disk Type: Ultra Disk |
| | | VPC ID: vpc-bi |

| Bestima tion | Parameter | Description |
|---------------------|----------------------|---|
| Instance Details | Database Type | Select Kafka. |
| | IP Address | Enter an IP address from the Default Endpoint parameter of the Message Queue for Apache Kafka instance. |
| | | Note To obtain an IP address, you can log on to the Message Queue for Apache Kafka console and navigate to the Instance Details page of the Message Queue for Apache Kafka instance. In the Basic Information section, you can obtain an IP address from the Default Endpoint parameter. |
| | Port Number | Enter the service port number of the Message Queue for Apache Kafka instance. The default port number is 9092. |
| | Database Account | Enter the username that is used to log on to the Message Queue for Apache Kafka instance. Note If the instance type of the Message Queue for Apache Kafka instance is VPC Instance, you do not need to specify the database account or database password. |
| | Database Password | Enter the password for the username. |
| | Topic | i. Click Get Topic List . ii. Select a topic name from the drop-down list. |
| | Kafka Version | Select a version based on the version of the Message Queue for Apache Kafka instance. |
| | Encryption | Select Non-encrypted or SCRAM-SHA-256 based on your business and security requirements. |

- 7. In the lower-right corner of the page, click **Set Whitelist and Next**.
- 8. Select the objects to be synchronized.



- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.



| Parameter | Description |
|----------------------------|---|
| Initial Synchronization | Select both Initial Schema Synchronization and Initial Full Data Synchronization. Before synchronizing incremental data, DTS synchronizes the schemas and historical data of the required objects from the source database to the destination database. |
| Filter options | Ignore DDL in incremental synchronization phase is selected by default. In this case, DTS does not synchronize DDL operations that are performed on the source database during incremental data synchronization. |

11. In the lower-right corner of the page, click **Precheck**.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.

On the Synchronization Tasks page, view the status of the data synchronization task.



7.10. Synchronize data from an ApsaraDB RDS for MySQL instance to a user-created Kafka cluster

Kafka is a distributed message queue service that features high throughput and high scalability. Kafka is widely used in big data fields such as log collection, monitoring data aggregation, streaming data processing, and online and offline analysis. It is indispensable for the big data ecosystem. This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to a user-created Kafka cluster by using Data Transmission Service (DTS). This allows you to extend message processing capabilities.

Prerequisites

- The version of the user-created Kafka cluster is 0.10 or 1.0.
- An ApsaraDB RDS for MySQL instance is created. For more information, see Create an ApsaraDB RDS for MySQL instance.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination database.

Limits

- You can select only tables as the objects to be synchronized.
- DTS does not synchronize the data in a renamed table to the destination Kafka cluster. This
 applies if the new table name is not included in the objects to be synchronized. To
 synchronize the data in a renamed table to the destination Kafka cluster, you must modify the
 objects to be synchronized. For more information, see Add an object to a data synchronization
 task.

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way many-to-one synchronization
- One-way cascade synchronization

Data format

The data that is synchronized to the Kafka cluster is stored in the Avro format. For more information, see DTS Avro schema.

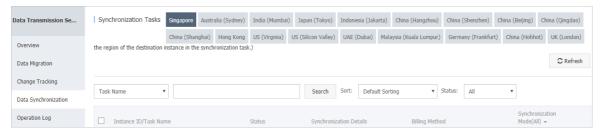
After the data is synchronized to the Kafka cluster, you must parse the data based on the Avro schema.

Billing

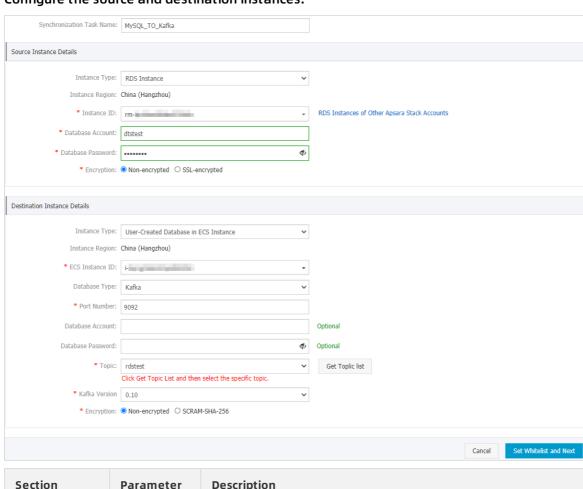
For more information, see Pricing.

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to MySQL, set Target Instance to Kafka, and set Synchronization Topology to One-Way Synchronization.

- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

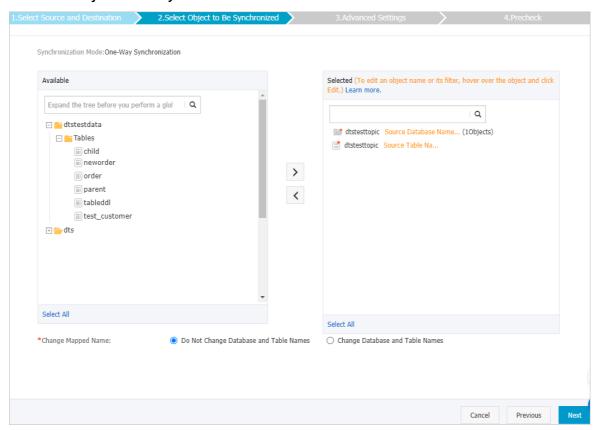


| Section | Parameter | Description |
|---------|----------------------------------|--|
| N/A | Synchroniza tion Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select RDS Instance. |

| Section | Parameter | Description |
|-------------------------------|----------------------|---|
| Source Instance Details | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the source RDS instance. |
| | Database Account | Enter the database account of the source RDS instance. The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. |
| | Database Password | Enter the password for the database account of the source RDS instance. |
| | Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. |
| | | Notice The Encryption parameter is available only for regions in mainland China and the Hong Kong (China) region. |
| | Instance Type | If the Kafka cluster is deployed on an ECS instance, select User-Created Database in ECS Instance |
| Destination | | If the Kafka cluster is deployed on a local server, select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway. |
| | | Note If you select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway, you must specify the VPC ID, IP address, and port number. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | ECS Instance ID | Select the ID of the ECS instance on which the Kafka cluster is deployed. |
| | Database Type | Select Kafka. |
| Instance Details | | |

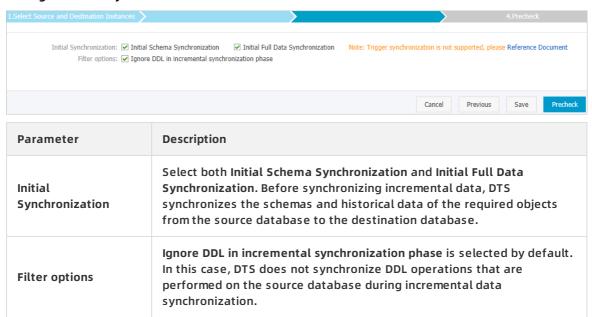
| Section | Parameter | Description |
|---------|----------------------|---|
| | Port Number | Enter the service port number of the Kafka cluster. The default port number is 9092. |
| | Database Account | Enter the username that is used to log on to the Kafka cluster. If the authentication is disabled for the Kafka cluster, you do not need to enter the username. |
| | Database Password | Enter the password for the username. If the authentication is disabled for the Kafka cluster, you do not need to enter the password. |
| | Topic | i. Click the Get Topic List. ii. Select a topic name from the drop-down list. |
| | Kafka Version | Select a version based on the version of the destination Kafka cluster. |
| | Encryption | Select Non-encrypted or SCRAM-SHA-256 based on your business and security requirements. |

- 7. In the lower-right corner of the page, click Set Whitelist and Next.
- 8. Select the objects to be synchronized.



| Parameter | Description | |
|----------------------------|--|--|
| | Select tables from the Available section and click the > icon to move the tables to the Selected section. | |
| Objects to be synchronized | Note DTS automatically maps the table names to the topic name that you select when you configure the source and destination instances. If you want to change the topic, you can move the pointer over the table and click Edit. You must specify a topic that exists in the Message Queue for Apache Kafka instance. | |

- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.

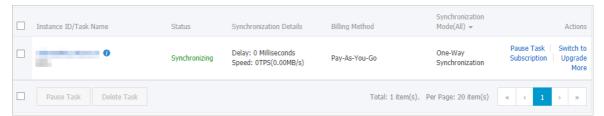


11. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed. Then, the data synchronization task starts.

You can view the status of the data synchronization task on the **Synchronization Tasks** page.



7.11. Synchronize data from a user-created MySQL database hosted on ECS to a user-created Kafka cluster

Kafka is a distributed message queue service that features high throughput and high scalability. Kafka is widely used for big data analytics such as log collection, data aggregation, streaming processing, and online and offline analysis. It is indispensable for the big data ecosystem. This topic describes how to synchronize data from a user-created MySQL database hosted on ECS to a user-created Kafka cluster by using Data Transmission Service (DTS). The data synchronization feature allows you to extend message processing capabilities.

Prerequisites

- The version of the user-created Kafka cluster is 0.10 or 1.0.
- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.

Limits

- You can select only tables as the objects to be synchronized.
- DTS does not synchronize the data in a renamed table to the destination Kafka cluster. This
 applies if the new table name is not included in the objects to be synchronized. To
 synchronize the data in a renamed table to the destination Kafka cluster, you must modify the
 objects to be synchronized. For more information, see Add an object to a data synchronization
 task.

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way many-to-one synchronization
- One-way cascade synchronization

Data format

The data that is synchronized to the Kafka cluster is stored in the Avro format. For more information, see DTS Avro schema.

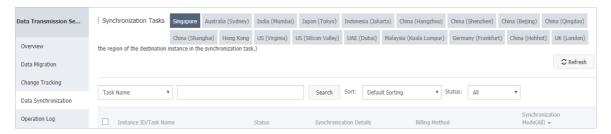
After the data is synchronized to the Kafka cluster, you must parse the data based on the Avro schema.

Preparations

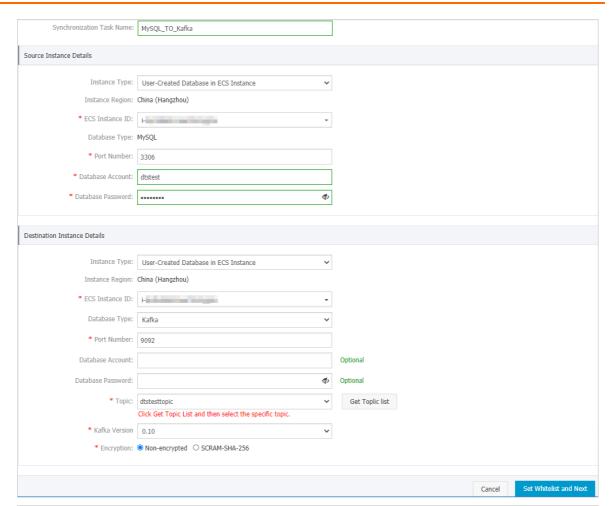
Before you configure a synchronization task, you must create an account and configure binary logging for the source instance. For more information, see Create an account for a user-created MySQL database and configure binary logging.

Procedure

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to MySQL, set Target Instance to Kafka, and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance, and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

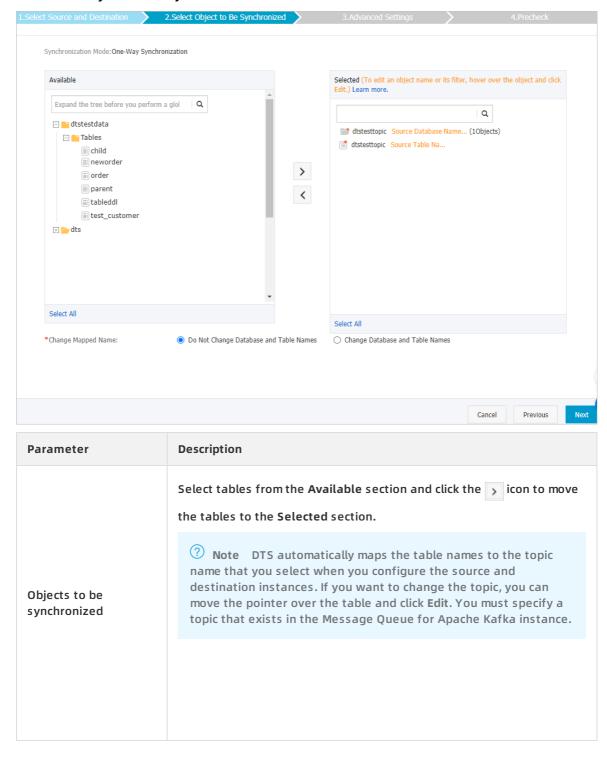


| Section | Parameter | Description |
|-------------------------------|----------------------------------|---|
| N/A | Synchroniza tion Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select User-Created Database in ECS Instance. |
| Source Instance Details | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | ECS Instance ID | Select the ID of the ECS instance that is connected to the user-created MySQL database. |
| | Database Type | The value of this parameter is set to MySQL and cannot be changed. |
| | Port Number | Enter the service port number of the user-created MySQL database. The default port number is 3306. |
| | | |

| Section | Parameter | Description |
|---------------------|----------------------|---|
| | Database Account | Enter the account of the user-created MySQL database. The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. |
| | Database Password | Enter the password of the source database account. |
| | Instance | If the Kafka cluster is deployed on an ECS instance, select User-Created Database in ECS Instance If the Kafka cluster is deployed on a local server, select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway. |
| | Туре | Note If you select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway, you must specify the VPC ID, IP address, and port number. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | ECS Instance ID | Select the ID of the ECS instance on which the Kafka cluster is deployed. |
| Destination | Database Type | Select Kafka. |
| Instance Details | Port Number | Enter the service port number of the Kafka cluster. The default port number is 9092. |
| | Database Account | Enter the username that is used to log on to the Kafka cluster. If the authentication is disabled for the Kafka cluster, you do not need to enter the username. |
| | Database Password | Enter the password of the username. If the authentication is disabled for the Kafka cluster, you do not need to enter the password. |
| | Topic | i. Click Get Topic List.ii. Select a topic name from the drop-down list. |
| | Kafka Version | Select the version of the destination Kafka cluster. |

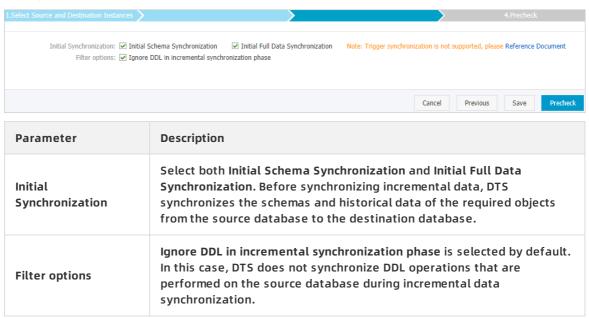
| Section | Parameter | Description |
|---------|------------|---|
| | Encryption | Select Non-encrypted or SCRAM-SHA-256 based on your business and security requirements. |

- 7. In the lower-right corner of the page, click Set Whitelist and Next.
- 8. Select the objects to be synchronized.

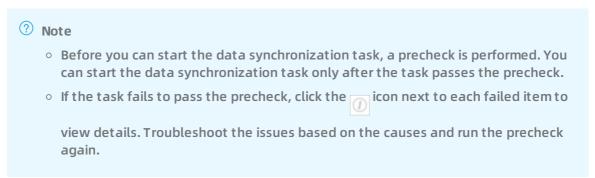


9. In the lower-right corner of the page, click Next.

10. Configure initial synchronization.

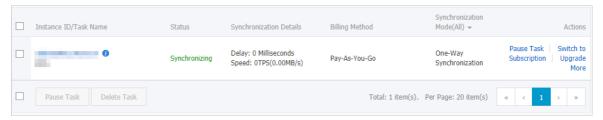


11. In the lower-right corner of the page, click Precheck.



12. Close the Precheck dialog box after the following message is displayed: The precheck is passed. Then, the data synchronization task starts.

You can view the status of the data synchronization task on the **Synchronization Tasks** page.



7.12. Synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to a user-created Kafka cluster

Kafka is a distributed message queue service that features high throughput and high scalability. Kafka is widely used for big data analytics such as log collection, data aggregation, streaming processing, and online and offline analysis. It is indispensable for the big data ecosystem. This topic describes how to synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to a user-created Kafka cluster by using Data Transmission Service (DTS). The data synchronization feature allows you to extend message processing capabilities.

Prerequisites

- The version of the user-created Kafka cluster is 0.10 or 1.0.
- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The user-created MySQL database is connected to Alibaba Cloud VPC over Express Connect, VPN Gateway, or Smart Access Gateway. For more information, see Connect to local IDCs.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.

Limits

- You can select only tables as the objects to be synchronized.
- DTS does not synchronize the data in a renamed table to the destination Kafka cluster. This
 applies if the new table name is not included in the objects to be synchronized. To
 synchronize the data in a renamed table to the destination Kafka cluster, you must modify the
 objects to be synchronized. For more information, see Add an object to a data synchronization
 task.

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way many-to-one synchronization
- One-way cascade synchronization

Data format

The data that is synchronized to the Kafka cluster is stored in the Avro format. For more information, see DTS Avro schema.

After the data is synchronized to the Kafka cluster, you must parse the data based on the Avro schema.

Billing

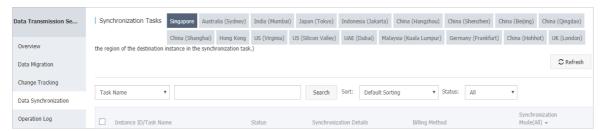
For more information, see Pricing.

Preparations

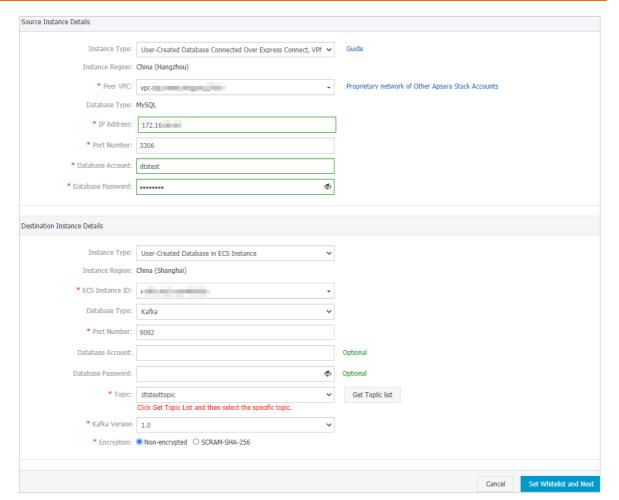
Before you configure a synchronization task, you must create an account and configure binary logging for the source instance. For more information, see Create an account for a user-created MySQL database and configure binary logging.

Procedure

- 1. Purchase a data synchronization instance.
 - Note On the buy page, set Source Instance to MySQL, set Target Instance to Kafka, and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance, and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

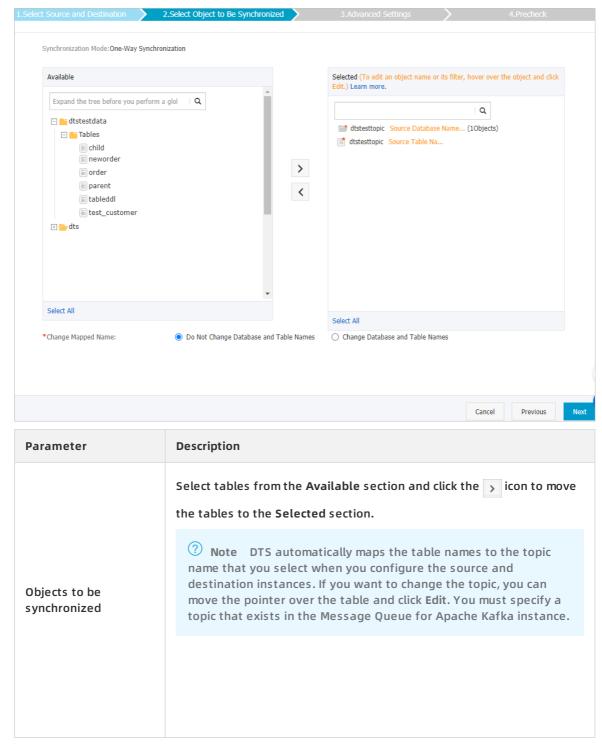


| Section | Parameter | Description |
|-------------------------------|----------------------------------|--|
| N/A | Synchroniza tion Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Peer VPC | Select the ID of the VPC that is connected to the user-created MySQL database. |
| Source Instance Details | Database Type | The value of this parameter is set to MySQL and cannot be changed. |
| | IP Address | Enter the server IP address of the user-created MySQL database. |
| | Port Number | Enter the service port number of the user-created MySQL database. The default port number is 3306. |

| Section | Parameter | Description |
|---------------------|----------------------|---|
| | Database Account | Enter the account of the user-created MySQL database. The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. |
| | Database Password | Enter the password of the source database account. |
| | Instance | If the Kafka cluster is deployed on an ECS instance, select User-Created Database in ECS Instance If the Kafka cluster is deployed on a local server, select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway. |
| | Туре | Note If you select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway, you must specify the VPC ID, IP address, and port number. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | ECS Instance ID | Select the ID of the ECS instance on which the Kafka cluster is deployed. |
| Destination | Database Type | Select Kafka. |
| Instance Details | Port Number | Enter the service port number of the Kafka cluster. The default port number is 9092. |
| | Database Account | Enter the username that is used to log on to the Kafka cluster. If the authentication is disabled for the Kafka cluster, you do not need to enter the username. |
| | Database Password | Enter the password of the username. If the authentication is disabled for the Kafka cluster, you do not need to enter the password. |
| | Topic | i. Click Get Topic List. ii. Select a topic name from the drop-down list. |
| | Kafka Version | Select the version of the destination Kafka cluster. |

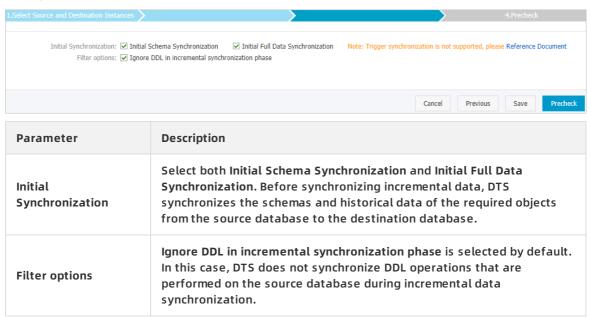
| Section | Parameter | Description |
|---------|------------|---|
| | Encryption | Select Non-encrypted or SCRAM-SHA-256 based on your business and security requirements. |

- 7. In the lower-right corner of the page, click Set Whitelist and Next.
- 8. Select the objects to be synchronized.



9. In the lower-right corner of the page, click Next.

10. Configure initial synchronization.

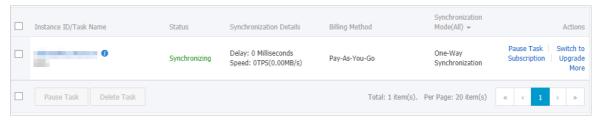


11. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to
 view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed. Then, the data synchronization task starts.

You can view the status of the data synchronization task on the **Synchronization Tasks** page.



8.Synchronize data to or from an Apsara PolarDB cluster

8.1. Configure two-way data synchronization between Apsara PolarDB for MySQL clusters

Data Transmission Service (DTS) supports two-way data synchronization between Apsara PolarDB for MySQL clusters. This feature is applicable to scenarios such as active georedundancy and geo-disaster recovery. This topic describes how to configure two-way data synchronization between Apsara PolarDB for MySQL clusters.

Prerequisites

- The source and destination Apsara PolarDB for MySQL clusters are created. For more information, see Create a PolarDB MySQL cluster.
- The binary logging feature is enabled for the source and destination Apsara PolarDB for MySQL clusters. For more information, see Enable binlogging.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source cluster.
- The source cluster must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination cluster may contain duplicate data records.

Limits

- DTS supports two-way data synchronization only between two PolarDB clusters. DTS does not support two-way data synchronization between multiple PolarDB clusters.
- If you have selected one or more tables (not a database) for synchronization, do not use ghost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.

Notice To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.

• Incompatibility with triggers

If the object to be synchronized is a database and the database contains a trigger that updates a synchronized table, data inconsistency may occur. For more information about how to solve this issue, see Configure a data synchronization task for a source database that contains a trigger.

• Limits on RENAME TABLE operations

RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if a table to be synchronized is renamed during data synchronization, the data of this table is not synchronized to the destination database. To avoid this situation, you can select the database to which this table belongs as the object when you configure the data synchronization task.

• Limits on DDL synchronization direction

To ensure the stability of a two-way synchronization channel, you can synchronize DDL updates of a single table only in one direction. If DDL synchronization in a direction is configured, DDL synchronization in the opposite direction is not supported. Only DML operations can be synchronized in the opposite direction.

SQL operations that can be synchronized

| Operati on type | SQL statements |
|--------------------|---|
| DML | INSERT, UPDATE, DELETE, and REPLACE |
| DDL | ALTER TABLE and ALTER VIEW CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW DROP INDEX and DROP TABLE RENAME TABLE TRUNCATE TABLE |

Conflict detection

To ensure data consistency, make sure that data records with the same primary key, business primary key, or unique key are updated only on one PolarDB cluster. If data records are updated on both two PolarDB clusters, a synchronization conflict occurs.

DTS checks and fixes conflicts to maximize the stability of two-way synchronization instances. DTS can detect the following types of conflicts:

• Uniqueness conflicts caused by INSERT operations

INSERT operations that do not comply with the uniqueness constraint cannot be synchronized. For example, if data records with the same primary key are inserted into two synchronization nodes at almost the same time, one of the inserted records fails to be synchronized. The synchronization fails because a record with the same primary key already exists in the other node.

- Inconsistent records caused by UPDATE operations
 - If the records to be updated do not exist in the destination cluster, DTS converts the UPDATE operation into an INSERT operation. However, uniqueness conflicts may occur.
 - The primary keys or unique keys of the records to be inserted may conflict with those of existing records in the destination instance.
- Non-existent records to be deleted

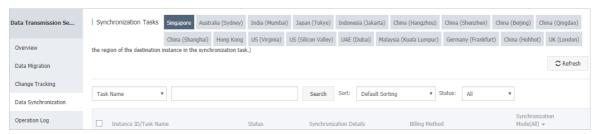
The records to be deleted do not exist in the destination cluster. In this case, DTS ignores the DELETE operation regardless of the conflict resolution policy that you have specified.



- During two-way synchronization, the system time of the source and destination instances may be different. Synchronization latency may occur. For these reasons, DTS cannot guarantee that the conflict detection mechanism can prevent all data conflicts. To perform two-way synchronization, make sure that records with the same primary key, business primary key, or unique key are updated only on one of the synchronization nodes.
- DTS provides conflict resolution policies to prevent conflicts that may occur during data synchronization. You can select a conflict resolution policy when you configure two-way data synchronization.

Procedure

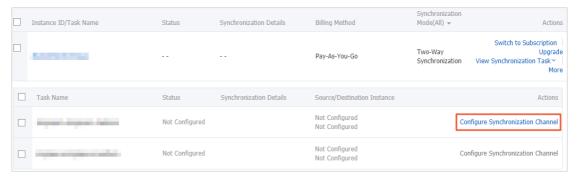
- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Notice On the buy page, set both Source Instance and Target Instance to PolarDB, and set Synchronization Topology to Two-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



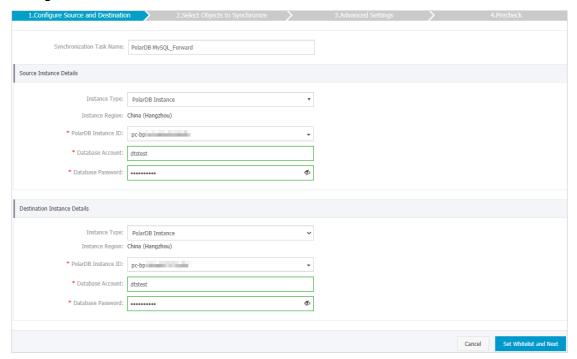
5. Configure the data synchronization task in one direction.

i. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column of the first data synchronization task.

Notice A two-way data synchronization instance contains two data synchronization tasks. You must configure a channel for each task. When you configure the second data synchronization task, find the task and click Configure Synchronization Channel in the Actions column.



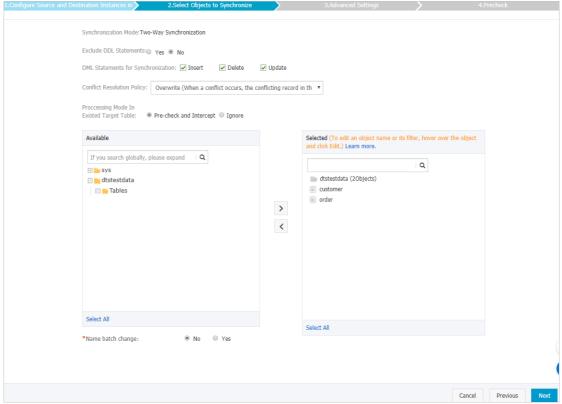
ii. Configure the source and destination instances.



| Section | Parameter | Description |
|---------|----------------------------------|--|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | The value of this parameter is set to PolarDB Instance and cannot be changed. |
| | | |

| Section | Parameter | Description |
|-------------------------------------|------------------------|---|
| Source Instance Details | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | PolarDB Instance ID | Select the ID of the source PolarDB cluster. Notice When you configure the data synchronization task in the opposite direction, select the ID of the destination PolarDB cluster. |
| | Database Account | Enter the database account of the source PolarDB cluster. The account must have the read/write permissions on the required objects. |
| | Database Password | Enter the password of the source database account. |
| | Instance Type | The value of this parameter is set to PolarDB Instance and cannot be changed. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | PolarDB Instance ID | Select the ID of the destination PolarDB cluster. |
| Destinatio n Instance Details | | Notice When you configure the data synchronization task in the opposite direction, select the ID of the source PolarDB cluster. |
| | Database Account | Enter the database account of the destination PolarDB cluster. The account must have the read/write permissions on the required objects. |
| | Database Password | Enter the password of the destination database account. |

- iii. In the lower-right corner of the page, click Set Whitelist and Next.
 DTS adds the CIDR blocks of DTS servers to the whitelists of the source and destination
 PolarDB clusters. This ensures that DTS servers can connect to the source and destination PolarDB clusters.
- iv. Configure the synchronization policy and objects.



| | | Cancel Previous Next |
|---------|---|---|
| Section | Parameter | Description |
| | Exclude DDL Statements | To exclude DDL operations, select Yes. To include DDL operations, select No. Notice After you select No, the table does not support synchronizing DDL operations in the opposite direction. |
| | DML Statements for Synchronizat ion | Select the type of DML operations to be synchronized. By default, the INSERT, UPDATE, and DELETE operations are selected. You can select the DML operation types based on your business requirements. |
| | | |

| Section | Parameter | Description |
|---------|----------------------------------|---|
| | Conflict Resolution Policy | Select the resolution policy in case of a synchronization conflict. By default, TaskFailed is selected. You can select a conflict resolution policy based on your business requirements. TaskFailed |
| | | The default conflict resolution policy. If a conflict occurs during data synchronization, the synchronization task reports an error and exits the process. The task enters a failed state and you must manually resolve the conflict. |
| | | If a conflict occurs during data synchronization, the synchronization task ignores the current statement and continues the process. The conflicting records in the destination database are used. |
| | | Overwrite |
| | | If a conflict occurs during data synchronization, the conflicting records in the destination database are overwritten. |

Synchronizat ion policy

| Section | Parameter | Description |
|---------|--|--|
| Section | Processing Mode In Existed Target Table | ■ Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the source and destination databases do not contain identical table names, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. Notice You can change the names of the tables to be synchronized by using the object name mapping feature. You can use this feature if the source and destination databases contain identical table names and tables in the destination database cannot be deleted or renamed. For more information, see Specify the name of an object in the destination instance. Ignore: skips the precheck for identical table names in the source and destination databases. Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. ■ DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. |
| | | |

| Section | Parameter | Description |
|----------------------------|-----------|--|
| Objects to be synchronized | N/A | Select objects (tables or a database) from the Available section and click the icon to move the objects to the Selected section. Notice If you select a database as the object to be |
| | | synchronized, all schema changes in the database are synchronized to the destination database. If you select a table as the object to be synchronized, only schema changes in this table are synchronized to the destination database. |
| | | After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the names of the objects that are synchronized to the destination database by using the object name mapping feature. For more information about how to use this feature, see Specify the name of an object in the destination instance. |

- v. In the lower-right corner of the page, click Next.
- vi. Configure initial synchronization.



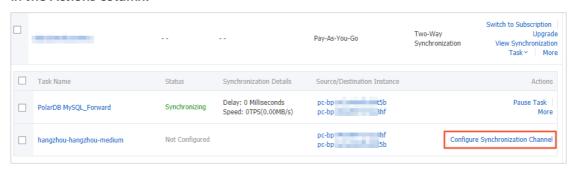
○ Notice

- During initial synchronization, DTS synchronizes the schemas and data of the required objects from the source cluster to the destination cluster. The schemas and data are the basis for subsequent incremental synchronization. Initial synchronization includes initial schema synchronization and initial full data synchronization. You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases.
- If all the required objects have been synchronized from the source cluster to the destination cluster, the data synchronization task in the opposite direction synchronizes only incremental data.

vii. In the lower-right corner of the page, click Precheck.



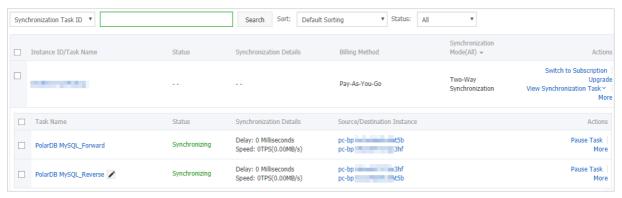
- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- viii. Close the Precheck dialog box after the following message is displayed: The precheck is passed. Then, the first data synchronization task starts.
- 6. Wait until the initial synchronization is complete and the first data synchronization task is in the Synchronizing state. You can view the status of the data synchronization task on the Synchronization Tasks page.
- 7. Configure the data synchronization task in the opposite direction.
 - i. Find the second data synchronization task, and click **Configure Synchronization Channel** in the Actions column.



ii. Repeat substeps ii to viii that are described in step 5.

Result

Wait until both data synchronization tasks are in the Synchronizing state.



8.2. Configure one-way data synchronization between Apsara PolarDB for MySQL clusters

Apsara PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is a high-performance, high-availability, easy-to-use, and reliable service that is compatible with the MySQL database engine. This topic describes how to configure one-way data synchronization between Apsara PolarDB for MySQL clusters.

Prerequisites

- The source and destination Apsara PolarDB for MySQL clusters are created. For more information, see Create an Apsara PolarDB for MySQL cluster.
- The binary logging feature is enabled for the source Apsara PolarDB for MySQL cluster. For more information, see Enable binary logging.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- If you have selected one or more tables (not a database) for synchronization, do not use ghost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.
 - Notice To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source instance.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination database.

SQL operations that can be synchronized

| O _I | perati n type | SQL statements |
|----------------|------------------|-------------------------------------|
| DI | ML | INSERT, UPDATE, DELETE, and REPLACE |

| Operati on type | SQL statements |
|--------------------|---|
| DDL | ALTER TABLE and ALTER VIEW CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW DROP INDEX and DROP TABLE RENAME TABLE TRUNCATE TABLE |

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization
- One-way many-to-one synchronization

For more information about synchronization topologies, see Synchronization topologies.

Limits

• Incompatibility with triggers

If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source database, the data in Table B becomes inconsistent between the source and destination databases.

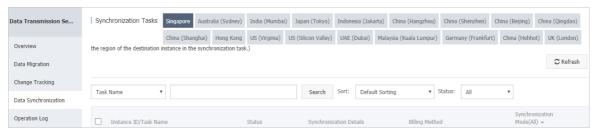
To avoid this situation, you must delete the trigger that is synchronized to the destination database and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.

Limits on RENAME TABLE operations

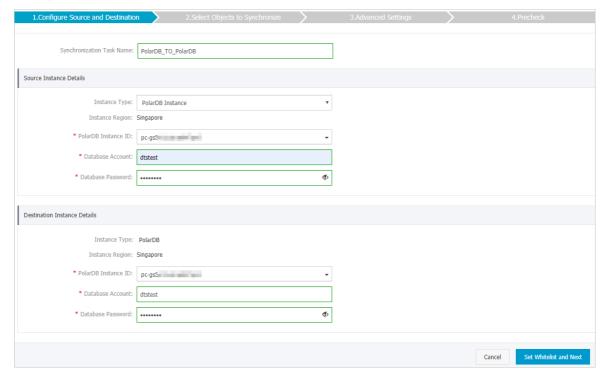
RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

Procedure

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set both Source Instance and Target Instance to POLARDB, and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



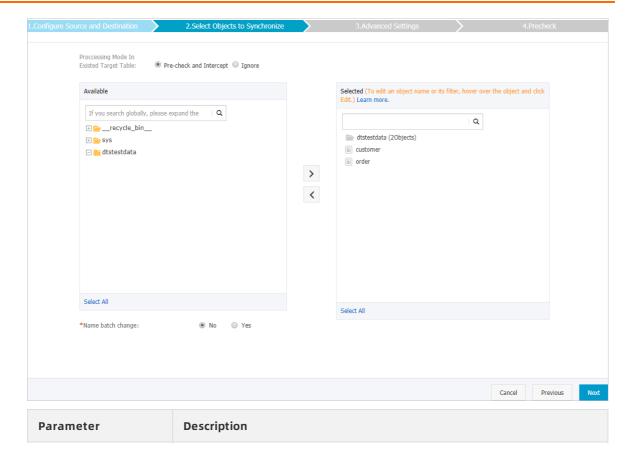
| Section | Parameter | Description |
|--------------------|----------------------------------|---|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | The value of this parameter is set to PolarDB Instance and cannot be changed. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | PolarDB Instance ID | Select the ID of the source PolarDB cluster. |
| Source Instance | | |

| Section | Parameter | Description |
|------------------------------------|------------------------|---|
| | Database Account | Enter the database account of the source PolarDB cluster. |
| | | ? Note The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. |
| | Database Password | Enter the password for the source database account. |
| | Instance Type | The value of this parameter is set to PolarDB and cannot be changed. |
| Destination Instance Details | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | PolarDB Instance ID | Select the ID of the destination PolarDB cluster. |
| | Database Account | Enter the database account of the destination PolarDB cluster. |
| | | Note The database account must have the ALL permission for the objects to be synchronized. |
| | Database Password | Enter the password for the destination database account. |

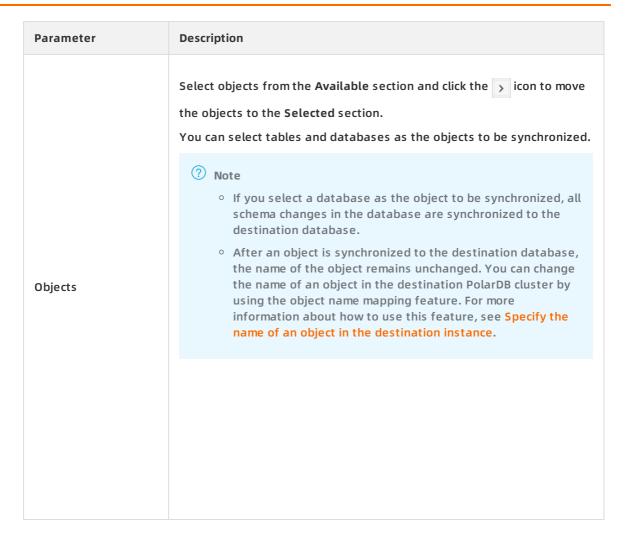
7. In the lower-right corner of the page, click Set Whitelist and Next.

? Note The CIDR blocks of DTS servers are automatically added to the whitelists of the source and destination PolarDB clusters. This ensures that DTS servers can connect to the source and destination PolarDB clusters.

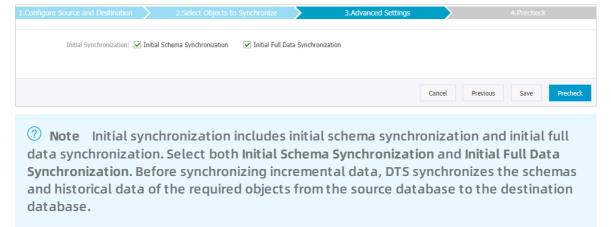
8. Configure the processing mode in existing destination tables and the objects to be synchronized.



| Description | |
|---|--|
| Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. | |
| Note If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. | |
| Ignore: skips the precheck for identical table names in the source and destination databases. | |
| Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. | |
| DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. | |
| If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. | |
| | |



- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.

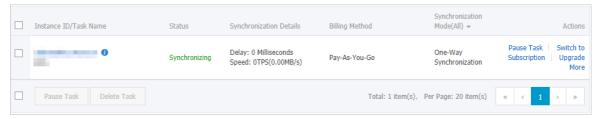


11. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.

On the Synchronization Tasks page, view the status of the data synchronization task.



8.3. Synchronize data between PolarDB clusters compatible with Oracle

Apsara PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is compatible with MySQL, PostgreSQL, and Oracle database engines. Apsara PolarDB provides superior performance in storage and computing to meet diverse requirements of enterprises. This topic describes how to synchronize data between PolarDB clusters compatible with Oracle by using Data Transmission Service (DTS).

Prerequisites

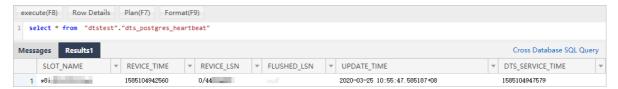
The tables to be synchronized contain primary keys or UNIQUE NOT NULL indexes.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- A single data synchronization task can synchronize data from only one database. To synchronize data from multiple databases, you must create a data synchronization task for

each database.

• To ensure that the delay time of data synchronization is accurate, DTS adds a heartbeat table named dts_postgres_heartbeat to the source database. The following figure shows the schema of the heartbeat table.



SQL operations that can be synchronized

- Data manipulation language (DML) operations: INSERT, UPDATE, and DELETE
- Data definition language (DDL) operations: CREATE TABLE (The CREATE TABLE operations to create partition tables or tables that contain functions cannot be synchronized.), ALTER TABLE, DROP TABLE, RENAME TABLE, CREATE INDEX, and ADD INDEX

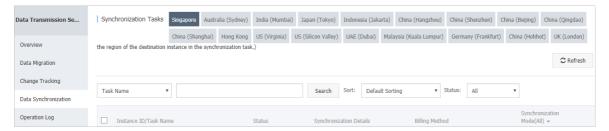
Preparation

Set the value of the wal_level parameter to *logical* for the source PolarDB cluster. This setting ensures that logical decoding is supported in write-ahead logging (WAL). For more information, see Change the values of cluster parameters.

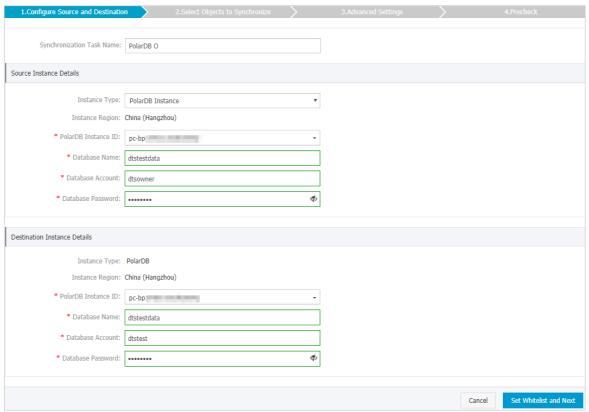
Warning If you change the value of the wal_level parameter, the source PolarDB cluster is restarted. Perform this operation with caution.

Procedure

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set both Source Instance and Target Instance to PolarDB, and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



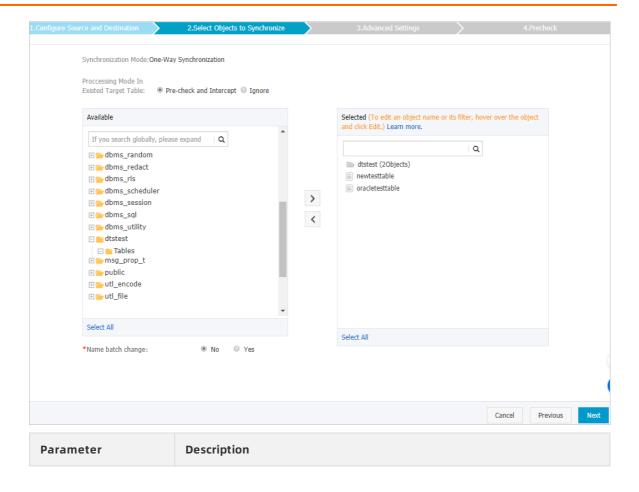
| Section | Parameter | Description |
|-------------------------------|----------------------------------|---|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | The value of this parameter is set to PolarDB Instance and cannot be changed. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | PolarDB Instance ID | Select the ID of the source PolarDB cluster. |
| Source Instance Details | Database Name | Enter the name of the source database. |
| | Database Account | Enter the privileged account of the source PolarDB cluster. For more information about how to create a privileged database account, see Create database accounts. |
| | Database Password | Enter the password of the source database account. |
| | | |

| Section | Parameter | Description |
|------------------------------------|------------------------|---|
| Destination Instance Details | Instance Type | The value of this parameter is set to PolarDB Instance and cannot be changed. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | PolarDB Instance ID | Select the ID of the destination PolarDB cluster. |
| | Database Name | Enter the name of the destination database. |
| | Database Account | Enter the database account of the destination PolarDB cluster. The account must have the owner permission on the database. |
| | | Notice You can specify the database owner when you create a database. |
| | Database Password | Enter the password of the destination database account. |

7. In the lower-right corner of the page, click Set Whitelist and Next.

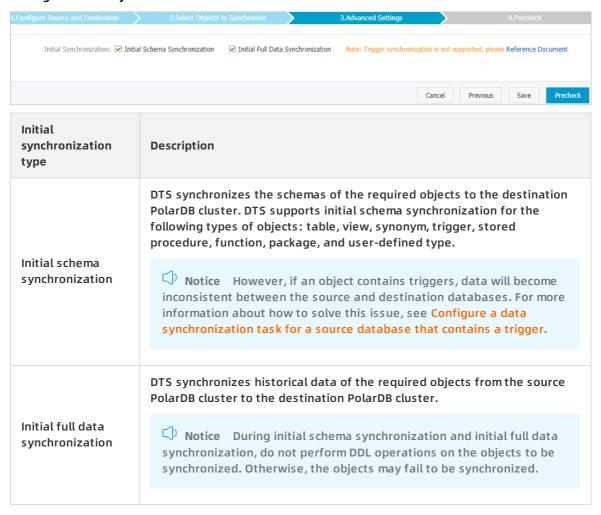
Note DTS adds the CIDR blocks of DTS servers to the whitelists of the source and destination PolarDB clusters. This ensures that DTS servers can connect to the source and destination PolarDB clusters.

8. Configure the processing mode in existing destination tables and the objects to be synchronized.



| Parameter | Description |
|-----------|---|
| | Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. Note If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. Ignore: skips the precheck for identical table names in the source and destination databases. DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data |
| Objects | Select objects from the Available section and click the icon to move the objects to the Selected section. You can select tables and databases as the objects to be synchronized. Note If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination PolarDB cluster by using the object name mapping feature. For more information about how to use this feature, see Specify the name of an object in the destination instance. |

- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.

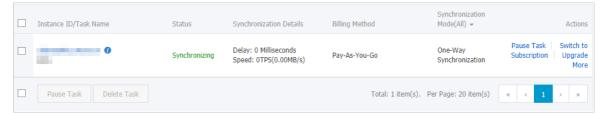


11. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You
 can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.

On the Synchronization Tasks page, view the status of the data synchronization task.



8.4. Synchronize data from a PolarDB cluster compatible with Oracle to a user-created Kafka cluster

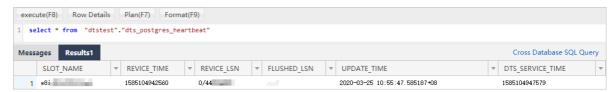
Kafka is a distributed message queue service that features high throughput and high scalability. Kafka is widely used for big data analytics such as log collection, data aggregation, streaming processing, and online and offline analysis. It is important for the big data ecosystem. This topic describes how to synchronize data from a PolarDB cluster compatible with Oracle to a user-created Kafka cluster by using Data Transmission Service (DTS). The data synchronization feature allows you to extend message processing capabilities.

Prerequisites

The tables to be synchronized contain primary keys or UNIQUE NOT NULL indexes.

Limits

- In this scenario, DTS supports only<dfn class="aliterm" aliterm-def="DTS synchronizes incremental data that is generated in the source database to the destination database."
 >incremental data synchronization</dfn>.DTS does not support and
- A single data synchronization task can synchronize data from only one database. To synchronize data from multiple databases, you must create a data synchronization task for each database.
- To ensure that the delay time of data synchronization is accurate, DTS adds a heartbeat table named dts_postgres_heartbeat to the source database. The following figure shows the schema of the heartbeat table.

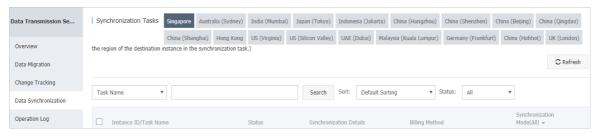


Preparation

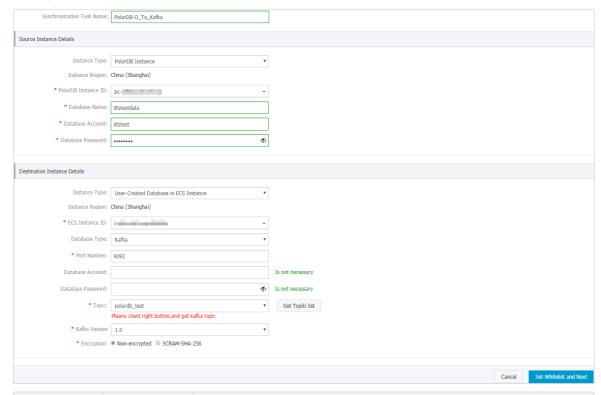
Set the value of the wal_level parameter to *logical* for the source PolarDB cluster. This setting ensures that logical decoding is supported in write-ahead logging (WAL). For more information, see Change the values of cluster parameters.

☐ Warning If you change the value of the wal_level parameter, the source PolarDB cluster is restarted. Perform this operation with caution.

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to PolarDB, set Target Instance to Kafka, and then set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



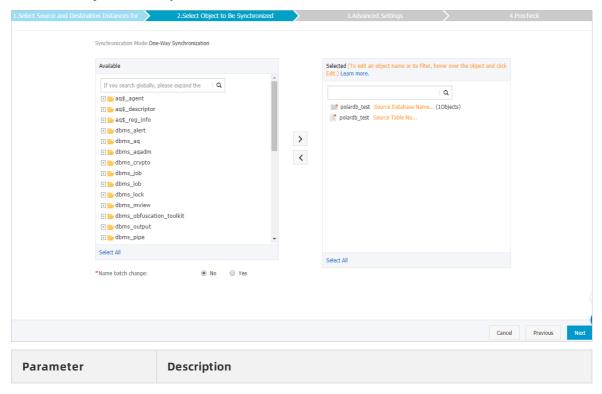
| Section | Parameter | Description |
|---------|----------------------------------|--|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |

| Section | Parameter | Description |
|---------------------|------------------------|---|
| Source Instance | Instance Type | The value of this parameter is set to PolarDB Instance and cannot be changed. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | PolarDB Instance ID | Select the ID of the source PolarDB cluster. |
| Details | Database Name | Enter the name of the source database. |
| | Database Account | Enter the privileged account of the source PolarDB cluster. For more information about how to create a privileged database account, see Create database accounts. |
| | Database Password | Enter the password of the source database account. |
| | Instance Type | Select an instance type based on where the user-created Kafka cluster is deployed. In this example, select User-Created Database with Public IP Address. |
| | | Note If you select other instance types, you must prepare the environment that is required for the database. For more information, see Preparation overview. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | ECS Instance ID | Select the ID of the ECS instance on which the Kafka cluster is deployed. |
| Destination | Database Type | Select Kafka. |
| Instance Details | Port Number | Enter the service port number of the Kafka cluster. The default port number is 9092. |
| | Database Account | Enter the username that is used to log on to the Kafka cluster. If authentication is disabled for the Kafka cluster, you do not need to enter the username. |
| | Database Password | Enter the password of the cluster account. If authentication is disabled for the Kafka cluster, you do not need to enter the password. |

| Section | Parameter | Description |
|---------|------------------|---|
| | Topic | Click Get Topic List , and select a topic name from the drop-down list. |
| | Kafka Version | Select the version of the user-created Kafka cluster. |
| | Encryption | Select Non-encrypted or SCRAM-SHA-256 based on your business and security requirements. |

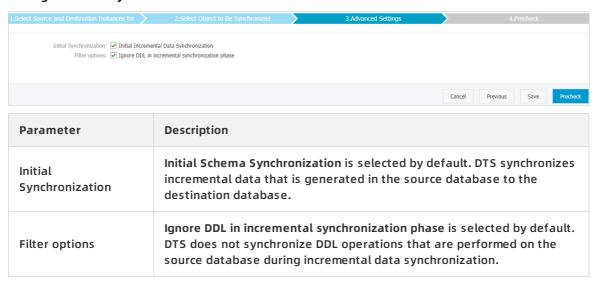
Note DTS adds the CIDR blocks of DTS servers to the whitelist of the source PolarDB cluster and the inbound rule of the destination ECS instance. This ensures that DTS servers can connect to the source cluster and the destination instance.

8. Select the objects to be synchronized.



| Select one or more tables from the Available section and click icon to move the tables to the Selected section. You can select tables as the objects to be synchronized. Objects Note DTS maps the table names to the topic name that select in step 6. If you want to change the topic name, you can select in step 6. | |
|---|-------------------------|
| | |
| the pointer over the table and click Edit. You must specify a that exists in the Kafka cluster. For more information, see Sy the name of an object in the destination instance. | u can move v a topic |

- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.

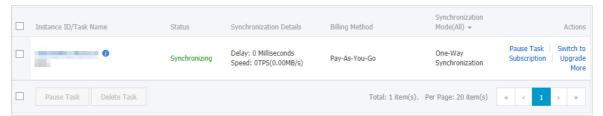


11. In the lower-right corner of the page, click **Precheck**.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the Synchronization Tasks page, view the status of the data synchronization task.



8.5. Synchronize data from an Apsara PolarDB for MySQL cluster to an ApsaraDB RDS for MySQL instance

This topic describes how to synchronize data from an Apsara PolarDB for MySQL cluster to an ApsaraDB RDS for MySQL instance by using Data Transmission Service (DTS).

Prerequisites

- An Apsara PolarDB for MySQL cluster is created. For more information, see Create an Apsara PolarDB for MySQL cluster.
- The binary logging feature is enabled for the Apsara PolarDB for MySQL cluster. For more information, see Enable binary logging.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- If you have selected one or more tables (not a database) for synchronization, do not use ghost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.
 - Notice To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source instance.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination database.

SQL operations that can be synchronized

| Operati on type | SQL statements |
|--------------------|---|
| DML | INSERT, UPDATE, DELETE, and REPLACE |
| DDL | ALTER TABLE and ALTER VIEW CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW DROP INDEX and DROP TABLE RENAME TABLE TRUNCATE TABLE |

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization
- One-way many-to-one synchronization

For more information about synchronization topologies, see Synchronization topologies.

Limits

• Incompatibility of triggers

If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source instance, the data in Table B becomes inconsistent between the source and destination instances.

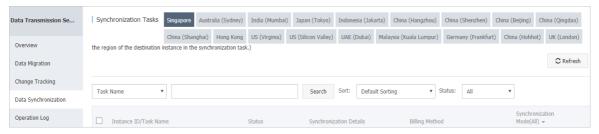
To avoid this situation, you must delete the trigger that is synchronized to the destination instance and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.

• Limits on RENAME TABLE operations

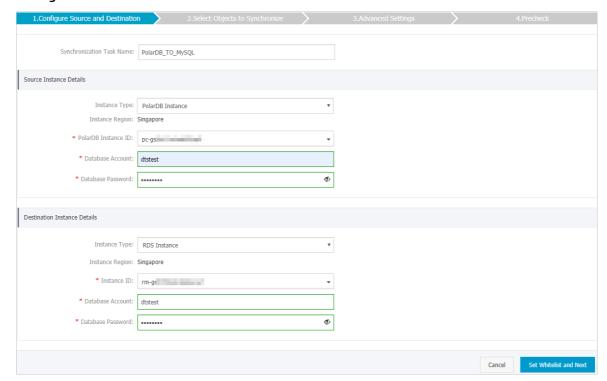
RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to PolarDB, Target Instance to MySQL, and Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.

- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

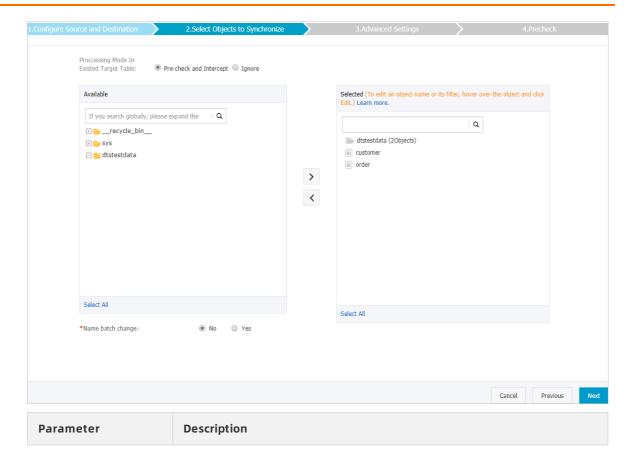


| Section | Parameter | Description |
|-------------------------------|----------------------------------|---|
| N/A | Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | The value of this parameter is set to PolarDB Instance and cannot be changed. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| Source Instance Details | PolarDB Instance ID | Select the ID of the source PolarDB cluster. |

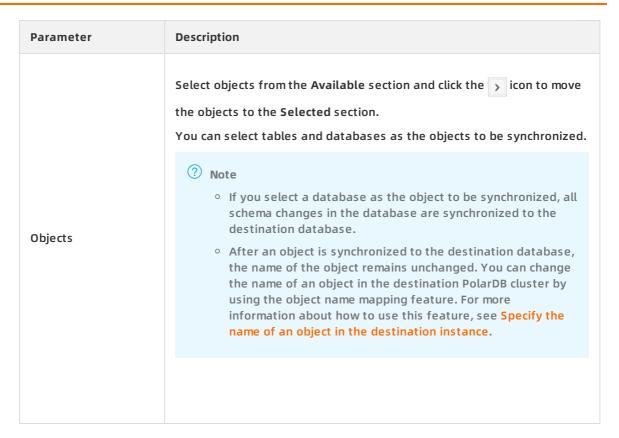
| Section | Parameter | Description |
|------------------------------------|----------------------|---|
| | Database Account | Enter the database account of the source PolarDB cluster. |
| | Database Password | Enter the password for the source database account. |
| | Instance Type | Select RDS instance. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Database Account | Enter the database account of the destination RDS instance. |
| | Database Password | Enter the password for the destination database account. |
| Destination Instance Details | | Note If the database engine of the destination RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password. |
| | | |
| | Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. |
| | | Note The Encryption parameter is available only in mainland China and Hong Kong(China). |
| | | |

Note The CIDR blocks of DTS servers are automatically added to the whitelists of the source PolarDB cluster and the destination RDS instance. This ensures that DTS servers can connect to the source PolarDB cluster and the destination RDS instance.

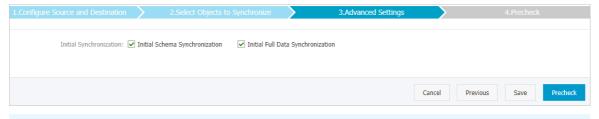
8. Configure the processing mode in existing destination tables and the objects to be synchronized.



| Parameter | Description |
|--|--|
| | Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. |
| | Note If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. |
| | Ignore: skips the precheck for identical table names in the source and destination databases. |
| Processing Mode In Existed Target Table | Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. |
| | |



- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.



Note Initial synchronization includes initial schema synchronization and initial full data synchronization. Select both Initial Schema Synchronization and Initial Full Data Synchronization. Before synchronizing incremental data, DTS synchronizes the schemas and historical data of the required objects from the source database to the destination database.

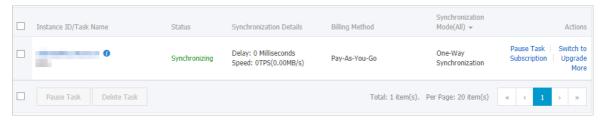
11. In the lower-right corner of the page, click Precheck.

? Note

- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.

On the Synchronization Tasks page, view the status of the data synchronization task.



8.6. Synchronize data from an Apsara PolarDB for MySQL cluster to an Elasticsearch instance

Alibaba Cloud Elasticsearch is compatible with open-source Elasticsearch features such as Security, Machine Learning, Graph, and Application Performance Management (APM). Alibaba Cloud Elasticsearch provides capabilities such as enterprise-level access control, security monitoring and alerts, and automatic report generation. You can use Alibaba Cloud Elasticsearch to search and analyze data. This topic describes how to synchronize data from an Apsara PolarDB for MySQL cluster to an Elasticsearch instance by using Data Transmission Service (DTS).

Prerequisites

- An Elasticsearch instance (5.5, 6.3, or 6.7 version) is created. For more information, see Create an Alibaba Cloud Elasticsearch instance.
- The binary logging feature for the Apsara PolarDB for MySQL cluster is enabled. For more information, see Enable-binary logging.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%
- DDL operations cannot be synchronized. If a DDL operation is performed on the table in the source database during data synchronization, you must perform the following steps: Remove the table from the required objects, remove the index for the table from the Elasticsearch instance, and then add the table to the required objects. For more information, see Remove an object from a data synchronization task and Add an object to a data synchronization task.

• To add columns to the table that you want to synchronize, perform the following steps: Modify the mappings of the table in the Elasticsearch instance, perform DDL operations in the Apsara PolarDB for MySQL cluster, and then pause and start the data synchronization task.

SQL operations supported by data synchronization

INSERT, DELETE, and UPDATE operations

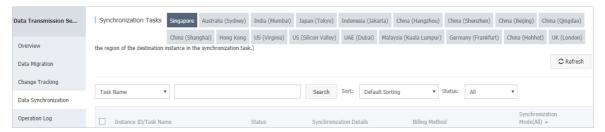
Data type mappings

The data types of the Apsara PolarDB for MySQL cluster and the Elasticsearch instance do not have one-to-one correspondence. During initial schema synchronization, DTS converts the data types of the Apsara PolarDB for MySQL cluster into those of the Elasticsearch instance. For more information, see Data type mappings for initial schema synchronization.

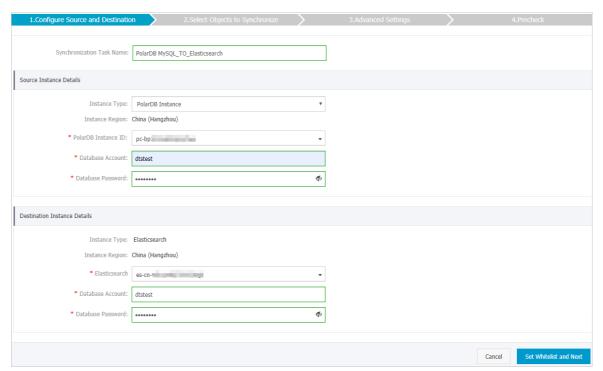
Procedure

(Optional)

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to PolarDB, set Target Instance to Elasticsearch, and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



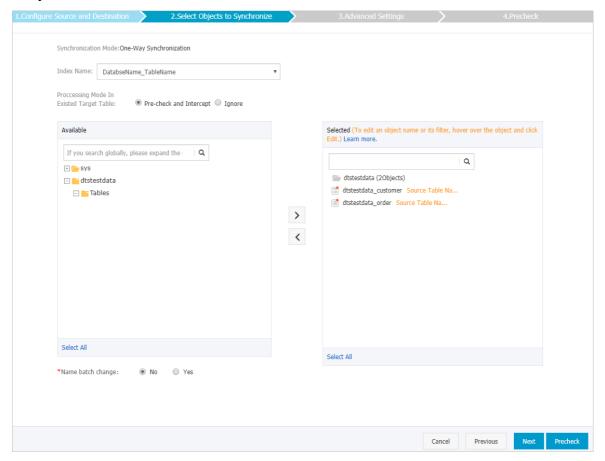
- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



| Section | Parameter | Description |
|-------------------------------|----------------------------------|---|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| Source Instance Details | Instance Type | The value of this parameter is set to PolarDB Instance and cannot be changed. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | PolarDB Instance ID | Select the ID of the source Apsara PolarDB for MySQL cluster. |
| | | Enter the database account of the Apsara PolarDB for MySQL cluster. |
| | Database Account | ? Note The account must have the read permission on the source database. |
| | Database Password | Enter the password of the source database account. |
| | Instance Type | The value of this parameter is set to Elasticsearch and cannot be changed. |

| Section | Parameter | Description |
|------------------------------------|----------------------|---|
| Destination Instance Details | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Elasticsearc h | Select the ID of the destination Elasticsearch instance. |
| | Database Account | Enter the account that is used to connect to the Elasticsearch instance. The default account is elastic. |
| | Database Password | Enter the password of the account. |

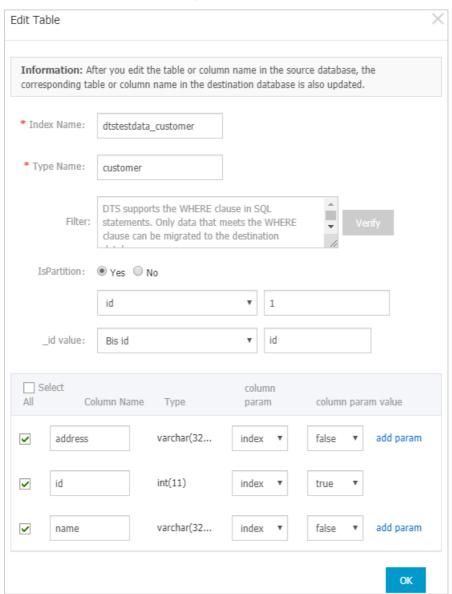
- 7. In the lower-right corner of the page, click Set Whitelist and Next.
 - ? Note DTS adds the CIDR blocks of DTS servers to the whitelists of the Apsara PolarDB for MySQL cluster and the Elasticsearch instance. This ensures that DTS servers can connect to the source cluster and the destination instance.
- 8. Configure the index name, the processing mode of identical index names, and the objects to be synchronized.



| Parameter | Description |
|--|--|
| Index Name | Table Name If you select Table Name, the name of the index that is created in the Elasticsearch instance is the same as the name of the table. In this example, the index name is customer. DatabaseName_TableName If you select DatabaseName_TableName, the name of the index that is created in the Elasticsearch instance is <database name="">_<table name="">. In this example, the index name is dtstestdata_customer. </table></database> |
| | Pre-check and Intercept: checks whether the destination database contains indexes that have the same names as tables in the source database. If the destination database does not contain indexes that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. |
| Processing Mode In Existed Target Table | Note If indexes in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. |
| | Ignore: skips the precheck for identical index names in the source and destination databases. |
| 3 | |
| | If the source and destination databases have the same mappings and the primary key of a record in the destination database is the same as that in the source database, the record remains unchanged during initial data synchronization. However, the record is overwritten during incremental data synchronization. |
| | If the source and destination databases have different mappings, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. |
| Ohierts to he | Select objects from the Available section and click the icon to move |
| Objects to be synchronized | the objects to the Selected section. You can select tables and databases as the objects to be synchronized. |

9. In the **Selected** section, move the pointer over a table, and then click **Edit**. In the Edit Table dialog box that appears, configure parameters for the table in the Elasticsearch instance,

such as the index name and type name.



| Parameter | Description |
|------------|---|
| Index Name | |
| | For more information, see Terms in the Elasticsearch documentation. |
| Type Name | |
| | |
| | |

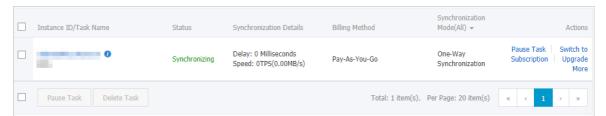
| Parameter | Description |
|-------------|--|
| Filter | Specify SQL conditions to filter data. Only data that meets the specified conditions is synchronized to the destination instance. For more information, see Use SQL conditions to filter data. |
| IsPartition | Select whether to set partitions. If you select Yes, you must also specify the partition key column and number of partitions. |
| _id value | Primary key column Composite primary key fields are merged into one column. Business key If you select a business key, you must also specify the business key column. |
| add param | Select the column parameter and parameter value. For more information, see Mapping parameters in the Elasticsearch documentation. |

10. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 11. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 12. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

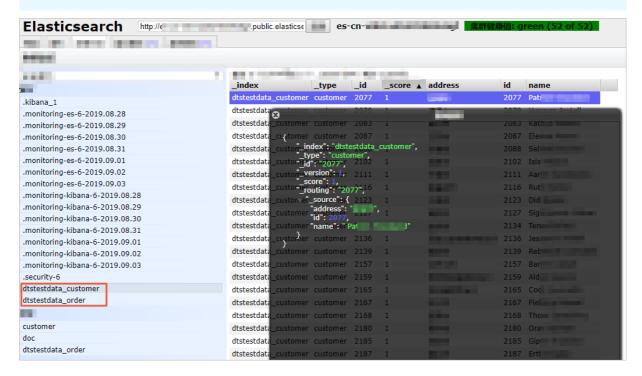
On the Synchronization Tasks page, view the status of the data synchronization task.



Check the index and data

If the data synchronization task is in the **Synchronizing** state, you can connect to the Elasticsearch instance and check whether the index is created and data is synchronized as expected. In this example, the **Elasticsearch Head** plug-in is used.

? Note If the index is not created or data is not synchronized as expected, you can delete the index and data, and then configure the data synchronization task again.



8.7. Synchronize data from an Apsara PolarDB for MySQL cluster to a user-created Kafka cluster

Kafka is a distributed message queue service that features high throughput and high scalability. Kafka is widely used for big data analytics such as log collection, data aggregation, streaming processing, and online and offline analysis. It is indispensable for the big data ecosystem. This topic describes how to synchronize data from an Apsara PolarDB for MySQL cluster to a user-created Kafka cluster by using Data Transmission Service (DTS). The data synchronization feature allows you to extend message processing capabilities.

Prerequisites

- The version of the user-created Kafka cluster is 0.10 or 1.0.
- A topic is created in the Kafka cluster to receive the synchronized data.
- The binary logging feature is enabled for the Apsara PolarDB for MySQL cluster. For more information, see Enable binary logging.

Precautions

The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.

Limits

- You can select only tables as the objects to be synchronized.
- DTS does not automatically update the objects to be synchronized based on their names.

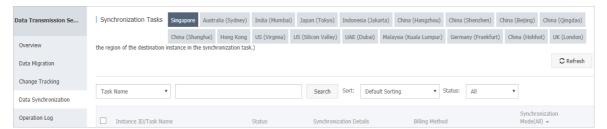
Note DTS does not synchronize the data in a renamed table to the destination Kafka cluster. This applies if the new table name is not included in the objects to be synchronized. To synchronize the data in a renamed table to the destination Kafka cluster, you must modify the objects to be synchronized. For more information, see Add an object to a data synchronization task.

Data format

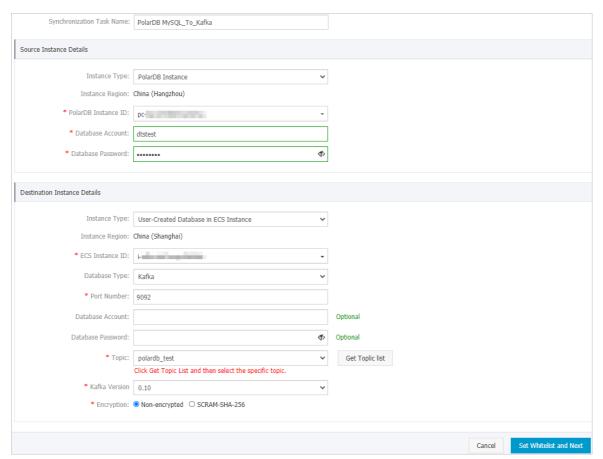
The data that is synchronized to the Kafka cluster is stored in the Avro format. For more information, see DTS Avro schema.

After the data is synchronized to the Kafka cluster, you must parse the data based on the Avro schema.

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to PolarDB, set Target Instance to Kafka, and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

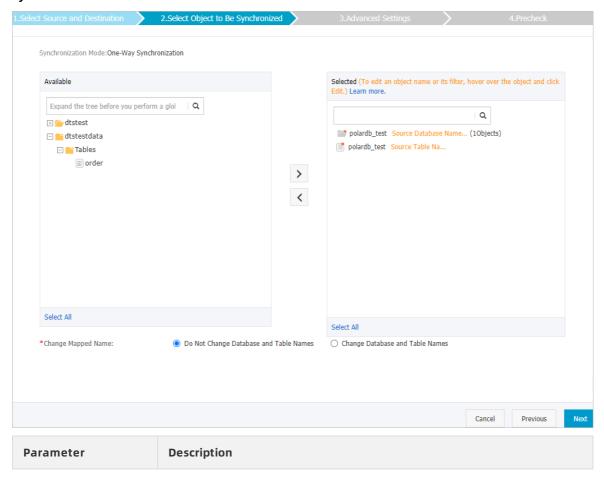


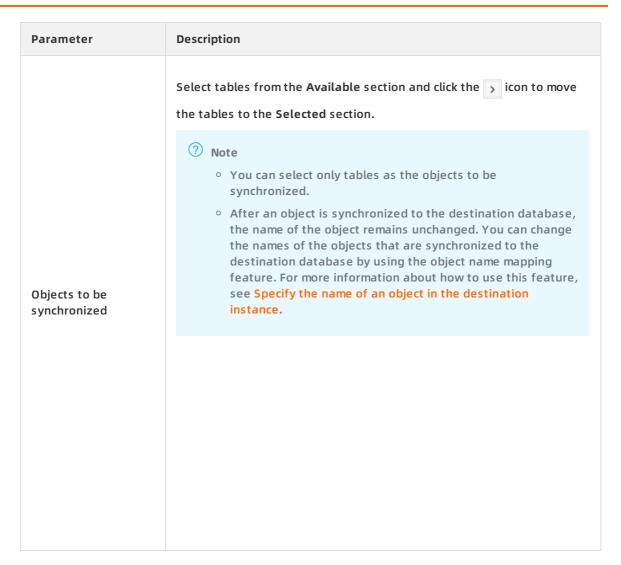
| Section | Parameter | Description |
|-------------------------------|----------------------------------|---|
| N/A | Synchroniza tion Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| Source Instance Details | Instance Type | The value of this parameter is set to PolarDB Instance and cannot be changed. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | PolarDB Instance ID | Select the ID of the source PolarDB cluster. |
| | Database Account | Enter the database account of the source PolarDB cluster. The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, and the permission to perform SELECT operations on the required objects. |
| | Database Password | Enter the password of the source database account. |
| | | |

| Section | Parameter | Description |
|------------------------------------|----------------------|---|
| | Instance Type | If the Kafka cluster is deployed on an ECS instance, select User-Created Database in ECS Instance If the Kafka cluster is deployed on a local server, select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway. Note If you select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway, you must specify the VPC ID, IP address, and port number. |
| | | The region of the destination instance. The region is the same |
| | Instance Region | as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | ECS Instance ID | Select the ID of the ECS instance on which the Kafka cluster is deployed. |
| Destination Instance Details | Database Type | Select Kafka. |
| | Port Number | Enter the service port number of the Kafka cluster. The default port number is 9092. |
| | Database Account | Enter the username that is used to log on to the Kafka cluster. If the authentication is disabled for the Kafka cluster, you do not need to enter the username. |
| | Database Password | Enter the password of the username. If the authentication is disabled for the Kafka cluster, you do not need to enter the password. |
| | Topic | i. Click Get Topic List . ii. Select a topic name from the drop-down list. |
| | Kafka Version | Select the version of the destination Kafka cluster. |
| | Encryption | Select Non-encrypted or SCRAM-SHA-256 based on your business and security requirements. |

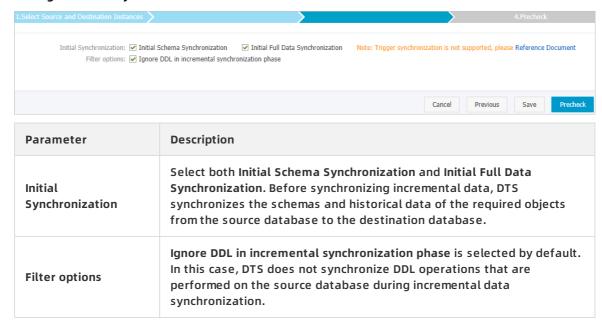
Note DTS adds the CIDR blocks of DTS servers to the whitelist of the source PolarDB cluster and the inbound rule of the destination ECS instance. This ensures that DTS servers can connect to the source cluster and destination instance.

8. Configure the processing mode in existing destination tables and the objects to be synchronized.





- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.

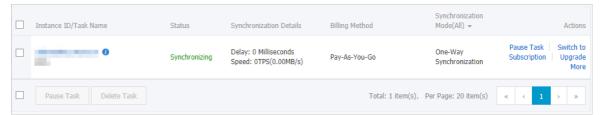


11. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.

On the Synchronization Tasks page, view the status of the data synchronization task.



8.8. Synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for MySQL cluster

AnalyticDB for MySQL is a real-time online analytical processing (RT-OLAP) service developed by Alibaba Cloud for online data analysis with high concurrency. AnalyticDB for MySQL can analyze petabytes of data from multiple dimensions at millisecond-level timing to provide you with data-driven insights into your business. This topic describes how to synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for MySQL cluster by using Data Transmission Service (DTS). AnalyticDB for MySQL allows you to build internal business intelligence (BI) systems, interactive query systems, and real-time report systems.

Prerequisites

- An AnalyticDB for MySQL cluster is created. For more information, see Create an AnalyticDB for MySQL cluster.
- The destination AnalyticDB for MySQL cluster has sufficient storage space.
- The binary logging feature is enabled for the Apsara PolarDB for MySQL cluster. For more information, see Enable binary logging.

Precautions

• DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is

unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.

- We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on objects during data synchronization. Otherwise, data synchronization may fail.
- If the disk space usage of nodes in an AnalyticDB for MySQL cluster reaches 80%, the cluster is locked. We recommend that you estimate the required disk space based on the objects to be synchronized. You must ensure that the destination cluster has sufficient storage space.

SQL operations that can be synchronized

- DDL operations: CREATE TABLE, DROP TABLE, RENAME TABLE, TRUNCATE TABLE, ADD COLUMN, and DROP COLUMN
- DML operations: INSERT, UPDATE, and DELETE

? Note If the data type of a field in the source table is changed during data synchronization, an error message is generated and the data synchronization task stops. You can submit a ticket or manually troubleshoot the issue. For more information, see Troubleshoot the synchronization failure that occurs due to field type changes.

Permissions required for database accounts

| Database | Required permission |
|--------------------------|---|
| Apsara PolarDB for MySQL | The read permission for the objects to be synchronized |
| AnalyticDB for MySQL | The read/write permissions for the objects to be synchronized |

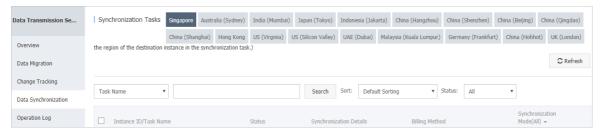
For more information about how to create and authorize a database account, see Create an Apsara PolarDB for MySQL database account and Create an AnalyticDB for MySQL database account.

Data type mapping

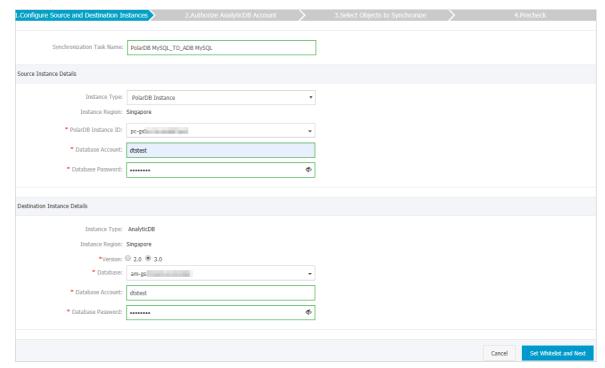
For more information, see Data type mappings for initial schema synchronization.

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to PolarDB, Target Instance to AnalyticDB for MySQL, and Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.

- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

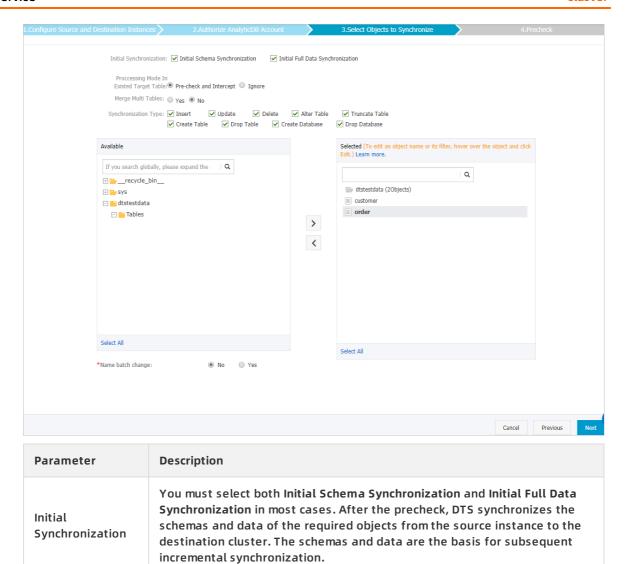


| Section | Parameter | Description |
|-------------------------------|-------------------------------|---|
| N/A | Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| Source Instance Details | Instance Type | The value of this parameter is set to PolarDB Instance and cannot be changed. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | PolarDB Instance ID | Select the ID of the source PolarDB cluster. |
| | | |

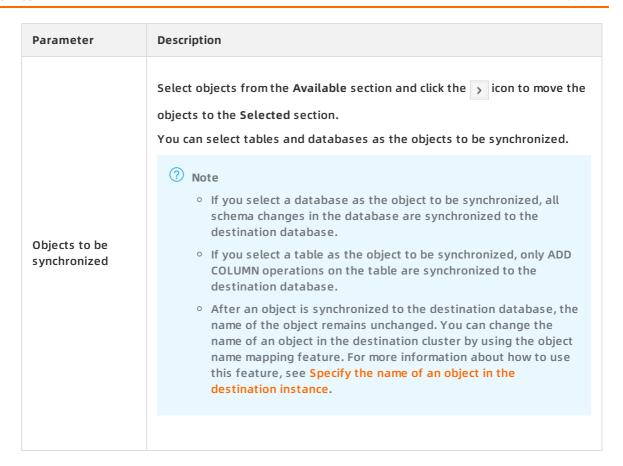
| Section | Parameter | Description |
|------------------------------------|----------------------|---|
| | Database Account | Enter the database account of the source PolarDB cluster. For more information about permissions required for the account, see Permissions required for database accounts. |
| | Database Password | Enter the password for the source database account. |
| Destination Instance Details | Instance Type | The value of this parameter is set to AnalyticDB and cannot be changed. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Version | Select 3.0. |
| | Database | Select the ID of the destination AnalyticDB for MySQL cluster. |
| | Database Account | Enter the database account of the AnalyticDB for MySQL cluster. For more information about permissions required for the account, see Permissions required for database accounts. |
| | Database Password | Enter the password for the destination database account. |

? Note The CIDR blocks of DTS servers are automatically added to the whitelists of the Apsara PolarDB for MySQL cluster and the AnalyticDB for MySQL cluster. This ensures that DTS servers can connect to the source and destination clusters.

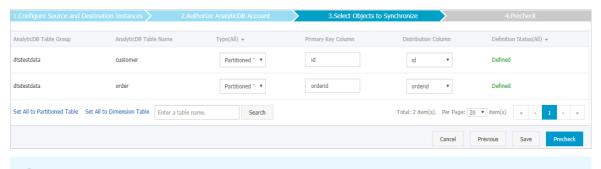
8. Configure the synchronization policy and objects.



| Parameter | Description | | |
|---|---|--|--|
| Processing Mode In Existed Target Table | Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. Note If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. Ignore: skips the precheck for identical table names in the source and destination databases. If the source and destination databases have the same schema, DTS does not synchronize data records that have the same primary keys as data records in the destination database. If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. | | |
| Merge Multi Tables | If you select Yes, DTS adds thedts_data_source column to each table to record data sources. In this case, DDL operations cannot be synchronized. No is selected by default. In this case, DDL operations can be synchronized. Note You can merge the data source columns based on tasks rather than tables. To merge only the data source columns of some tables, you can create two data synchronization tasks. | | |
| Synchronization Type | Select the types of operations that you want to synchronize based on your business requirements. All operation types are selected by default. ? Note Only INSERT, UPDATE, DELETE, and ADD COLUMN operations can be synchronized. | | |



- 9. In the lower-right corner of the page, click Next.
- 10. Specify a type for the tables to be synchronized to the destination database.



Note After you select Initial Schema Synchronization, you must specify the type, primary key column, and partition key column for the tables to be synchronized to AnalyticDB for MySQL. For more information, see CREATE TABLE.

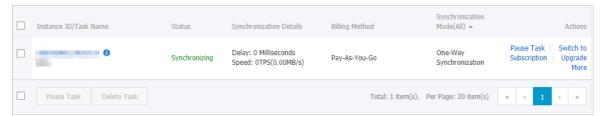
11. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to
 view details. Troubleshoot the issues based on the causes and run the precheck again.

- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the Synchronization Tasks page, view the status of the data synchronization task.



8.9. Synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for PostgreSQL instance

AnalyticDB for PostgreSQL (previously known as HybridDB for PostgreSQL) is a fast, easy-to-use, and cost-effective warehousing service. AnalyticDB for PostgreSQL supports processing petabytes of data. This topic describes how to synchronize data from an Apsara PolarDB for MySQL cluster to an AnalyticDB for PostgreSQL instance by using Data Transmission Service (DTS). This is applicable to scenarios such as ad-hoc query and analysis, extract, transform, and load (ETL) operations, and data visualization.

Prerequisites

- The binary logging feature for the Apsara PolarDB for MySQL cluster is enabled. For more information, see Enable binlogging.
- The tables to be synchronized from the Apsara PolarDB for MySQL cluster contain primary keys.
- An AnalyticDB for PostgreSQL instance is created. For more information, see Create an instance.

Notes

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- During initial full data synchronization, concurrent INSERT operations cause segments in the tables of the destination instance. After initial full data synchronization, the tablespace of the destination instance is larger than that of the source cluster.

Limits

- You can select only tables as the objects to be synchronized.
- You cannot synchronize the following types of data: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID SNAPSHOT.
- We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on objects during data synchronization. Otherwise, data synchronization may fail.

Supported SQL operations

- DML operations: INSERT, UPDATE, and DELETE
- DDL operations: ADD COLUMN, and RENAME COLUMN

Note The CREATE TABLE operation is not supported. To synchronize data from a new table, you must add the table to the selected objects. For more information, see Add an object to a data synchronization task.

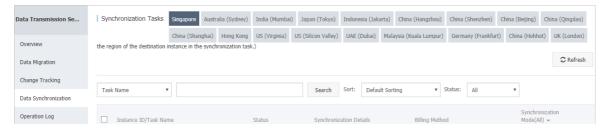
Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way many-to-one synchronization

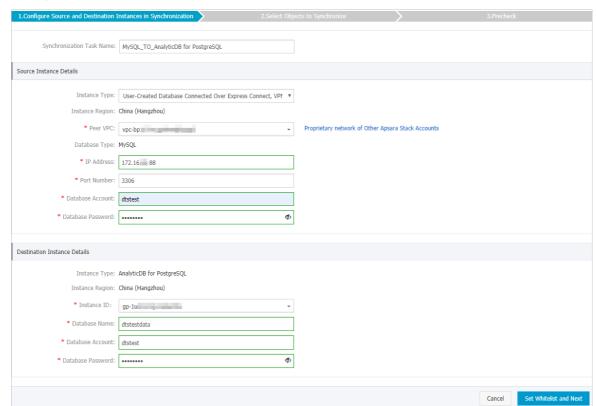
Term mappings

| Term in Apsara PolarDB for MySQL | Term in AnalyticDB for PostgreSQL |
|----------------------------------|-----------------------------------|
| Database | Schema |
| Table | Table |

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the purchase page, select MySQL for the source instance and AnalyticDB for PostgreSQL for the destination instance. Select One-Way Synchronization as the synchronization topology.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



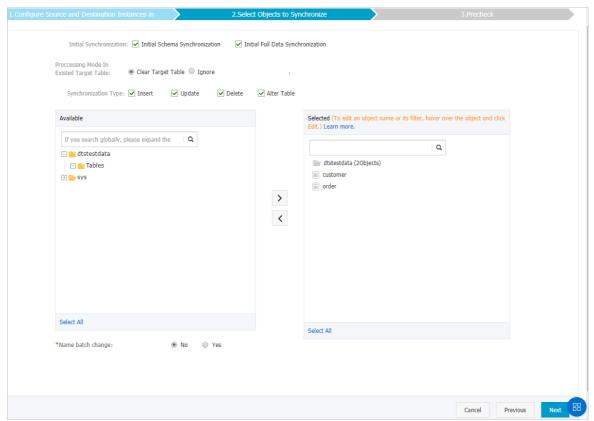
| Section | Parameter | Description |
|---------|--|---|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | | Select User-Created Database Connected Over Express Connect, VPN Gateway, or Smart Access Gateway. |
| | Instance Type Instance Region | Note You cannot select Apsara PolarDB for MySQL cluster as the instance type. To synchronize data from a Apsara PolarDB for MySQL cluster, you can select User-Created Database Connected Over Express Connect, VPN Gateway, or Smart Access Gateway. |
| | | The region of the Apsara PolarDB for MySQL cluster. The value is the same as that you selected when purchasing the data synchronization instance. You cannot change the value of this parameter. |

| Section | Parameter | Description |
|--------------------|----------------------|--|
| | Peer VPC | Select the ID of the VPC where the Apsara PolarDB for MySQL cluster resides. To obtain the VPC ID, you can log on to the Apsara PolarDB console and click the cluster ID. On the Overview page that appears, you can view the ID of the VPC where the cluster resides in the Basic Information section. Consolies Information Section. Consolies Information Section. Consolies Information Basic Information Consolies ID Cons |
| Source Instance | Database Type | The value of this parameter is set to MySQL and cannot be changed. |
| Details | IP Address | Enter the private IP address of the Apsara PolarDB for MySQL cluster. In this example, enter 172.16.20.36. You can obtain the private IP address by pinging the VPC-facing endpoint of the Apsara PolarDB for MySQL cluster. Connection Information © Viniteists © Create Viniteists Oreale Viniteists Primary Endpoints Primary Endpoint Apply Node Settings |
| | Port Number | Enter the port number of the Apsara PolarDB for MySQL cluster. The default port number is 3306. |
| | Database Account | Enter the database account for the Apsara PolarDB for MySQL cluster. Note The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. |
| | Database Password | Enter the password for the database account. |

| Section | Parameter | Description |
|------------------------------------|----------------------|--|
| | Instance Type | The value of this parameter is set to AnalyticDB for PostgreSQL and cannot be changed. |
| | Instance Region | The region of the destination instance. The value is the same as that you selected when purchasing the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the destination AnalyticDB for PostgreSQL instance. |
| Destination Instance Details | Database Name | Enter the name of the destination database in the AnalyticDB for PostgreSQL instance. |
| Details | Database Account | Enter the database account for the destination AnalyticDB for PostgreSQL instance. |
| | | ? Note The database account must have the ALL permission for the objects to be synchronized. |
| | Database Password | Enter the password for the database account. |

? Note The CIDR blocks of DTS servers are automatically added to the whitelists of the Apsara PolarDB for MySQL cluster and the AnalyticDB for PostgreSQL instance. This ensures that DTS servers can connect to the source cluster and destination instance.

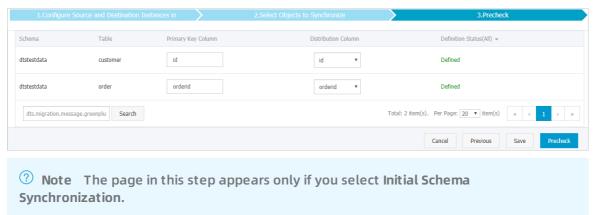
8. Configure the synchronization policy and objects.



| Initial Synchronization Initial Synchronization in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization. Initial Synchronization Initial Synchronization in most cases. After the precheck, DTS synchronize the schemas and data of the extination that synchronization. In subsequent incremental synchronization in table before initial full data synchronization. If you want to synchronization task, you can select this mode. Ignore Skips the Schema Name Conflict item during the precheck. Adds new data to the existing data during initial full data synchronization. You can select this mode if you want to synchronize data from multiple tables to one table. | Section | Parameter | Description |
|---|---------|-------------------------|--|
| Skips the Schema Name Conflict item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. Synchronizat ion policy Skips the Schema Name Conflict item during the precheck. Adds new data to the existing data during initial full data synchronization. You can select this mode if you want to synchronize data | | Initial Synchronization | and Initial Full Data Synchronization in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental |
| ion policy during initial full data synchronization. You can select this mode if you want to synchronize data | _ | _ | Skips the Schema Name Conflict item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. • Ignore Skips the Schema Name Conflict item during the |
| | | | during initial full data synchronization. You can select this mode if you want to synchronize data |

| Section | Parameter | Description |
|-------------------------------|----------------------|---|
| | Synchronization Type | Select the types of operations that you want to synchronize based on your business requirements. Insert Update Delete Alter Table |
| Objects to be synchronized | | Select tables from the Available section and click the right arrow (>) icon to add the tables to the Selected section. |
| | N/A | Note You can select only tables as the objects to be synchronized. You can change the names of columns in the destination database by using the object name mapping feature provided by DTS. For more information about how to use this feature, see Specify the name of an object in the destination instance. |
| | | |

9. Specify the primary key column and distribution column of the table that you want to synchronize to the AnalyticDB for PostgreSQL instance.

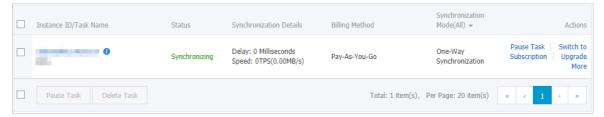


10. In the lower-right corner of the page, click **Precheck**.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 11. Close the Precheck dialog box after the following message is displayed: The precheck is passed. Then, the data synchronization task starts.
- 12. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.

On the Synchronization Tasks page, view the status of the data synchronization task.



8.10. Synchronize data from a user-created MySQL database hosted on ECS to an Apsara PolarDB for MySQL cluster

Apsara PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is a high-performance, high-availability, easy-to-use, and reliable service that is compatible with the MySQL database engine. This topic describes how to synchronize data from a user-created MySQL database hosted on ECS to an Apsara PolarDB for MySQL cluster by using Data Transmission Service (DTS).

Prerequisites

An Apsara PolarDB for MySQL cluster is created. For more information, see Create an Apsara PolarDB for MySQL cluster.

Precautions

• DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less

than 30%.

• If you have selected one or more tables (not a database) for synchronization, do not use ghost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.

Notice To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.

- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source instance.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination database.

SQL operations that can be synchronized

| Operati on type | SQL statements |
|--------------------|---|
| DML | INSERT, UPDATE, DELETE, and REPLACE |
| DDL | ALTER TABLE and ALTER VIEW CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW DROP INDEX and DROP TABLE RENAME TABLE TRUNCATE TABLE |

Limits

· Incompatibility with triggers

If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source database, the data in Table B becomes inconsistent between the source and destination databases.

To avoid this situation, you must delete the trigger that is synchronized to the destination database and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.

• Limits on RENAME TABLE operations

RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

Preparations

Create an account for a user-created MySQL database and configure binary logging

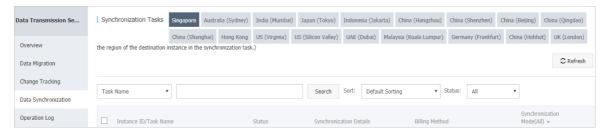
Note The database accounts must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.

Supported synchronization topologies

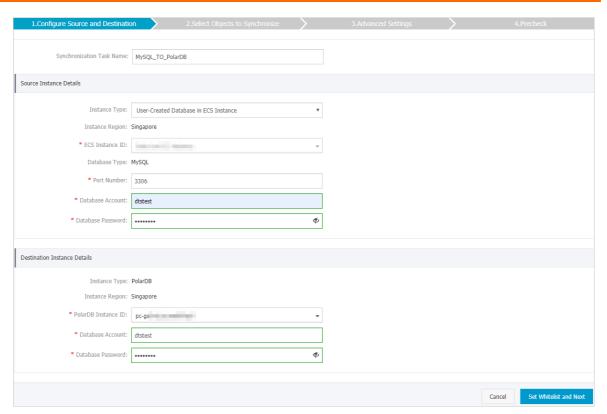
- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization
- One-way many-to-one synchronization

For more information about synchronization topologies, see Synchronization topologies.

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to MySQL, Target Instance to PolarDB, and Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



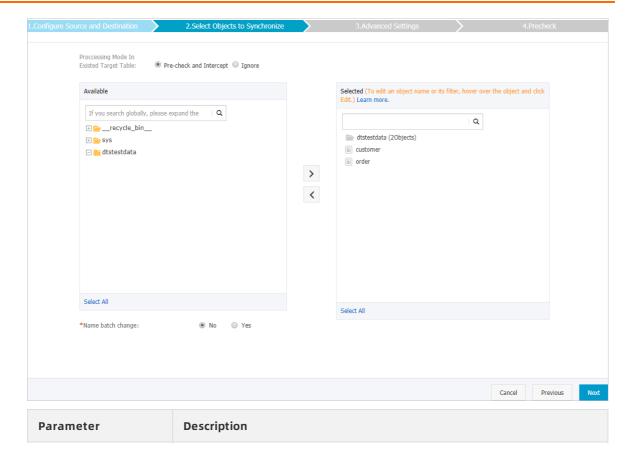
| Section | Parameter | Description |
|-------------------------------|----------------------------------|--|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select User-Created Database in ECS Instance. |
| Source Instance Details | Instance Region | The region of the source instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | ECS Instance ID | Select the ID of the ECS instance that is connected to the user-created MySQL database. |
| | Database Type | The value of this parameter is set to MySQL and cannot be changed. |
| | Port Number | Enter the service port number of the user-created MySQL database. |
| | Database Account | Enter the account of the user-created MySQL database. |
| | | Note The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects. |

| Section | Parameter | Description |
|------------------------------------|------------------------|---|
| | Database Password | Enter the password for the account of the user-created MySQL database. |
| | Instance Type | The value of this parameter is set to PolarDB and cannot be changed. |
| Destination Instance Details | Instance Region | The region of the destination instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | PolarDB Instance ID | The ID of the destination PolarDB cluster. |
| | Database Account | Enter the database account of the destination PolarDB cluster. |
| | | ? Note The database account must have the ALL permission for the objects to be synchronized. |
| | | |
| | Database Password | Enter the password for the database account. |

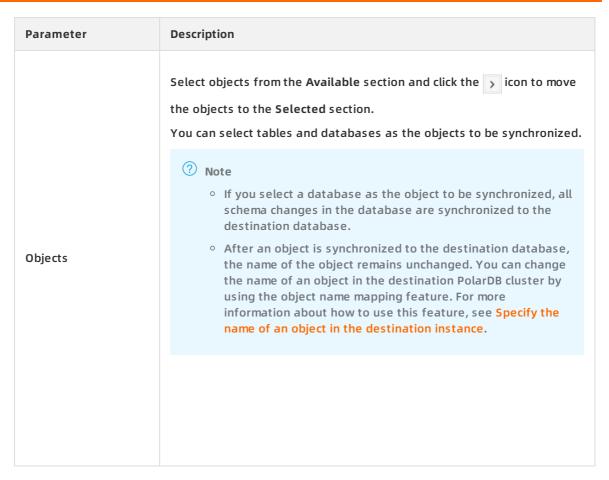
7. In the lower-right corner of the page, click Set Whitelist and Next.

Note The CIDR blocks of DTS servers are automatically added to the inbound rule of the ECS instance and the whitelist of the destination PolarDB cluster. This ensures that DTS servers can connect to the source instance and the destination PolarDB cluster.

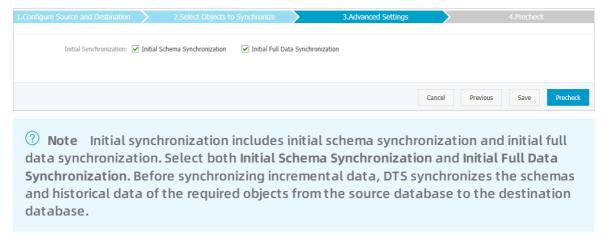
8. Configure the processing mode in existing destination tables and the objects to be synchronized.



| Parameter | Description | |
|--|--|--|
| | Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. | |
| | Note If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. | |
| | Ignore: skips the precheck for identical table names in the source and destination databases. | |
| Processing Mode In Existed Target Table | ✓ Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. ■ DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. ■ If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. | |



- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.

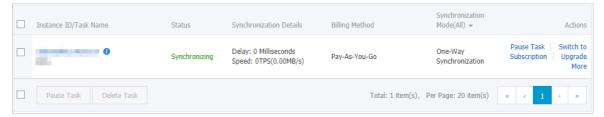


11. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.

On the Synchronization Tasks page, view the status of the data synchronization task.



8.11. Synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an Apsara PolarDB for MySQL cluster

Apsara PolarDB is a next-generation relational database service developed by Alibaba Cloud. It is a high-performance, high-availability, easy-to-use, and reliable service that is compatible with the MySQL database engine. This topic describes how to synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an Apsara PolarDB for MySQL cluster by using Data Transmission Service (DTS).

Prerequisites

- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The user-created MySQL database is connected to Alibaba Cloud VPC over Express Connect, VPN Gateway, or Smart Access Gateway. For more information, see Connect to local IDCs.

Note DTS is allowed to access the VPC to which the user-created MySQL database belongs. For more information, see Configure a route between DTS and Express Connect, VPN Gateway, or Smart Access Gateway. • An Apsara PolarDB for MySQL cluster is created. For more information, see Create an Apsara PolarDB for MySQL cluster.

Note The available storage space of the destination ApsaraDB RDS for MySQL database is larger than the total size of the data in the user-created MySQL database.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- If you have selected one or more tables (not a database) for synchronization, do not use ghost or pt-online-schema-change to modify the tables during data synchronization. Otherwise, data synchronization may fail.

Notice To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see Change the table schema without locking.

- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source database.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination cluster.

SQL operations that can be synchronized

| Operati on type | SQL statements |
|--------------------|---|
| DML | INSERT, UPDATE, DELETE, and REPLACE |
| DDL | ALTER TABLE and ALTER VIEW CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW DROP INDEX and DROP TABLE RENAME TABLE TRUNCATE TABLE |

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization

- One-way cascade synchronization
- One-way many-to-one synchronization

For more information about synchronization topologies, see Synchronization topologies.

Limits

Incompatibility with triggers

If the object you want to synchronize is a database and the database contains a trigger that updates the synchronized table, the synchronized data may be inconsistent. For example, the source database contains Table A and Table B. If a data record is inserted into Table A, a trigger inserts a data record into Table B. In this case, after an INSERT operation is performed on Table A in the source database, the data in Table B becomes inconsistent between the source and destination databases.

To avoid this situation, you must delete the trigger that is synchronized to the destination database and select Table B as the object to be synchronized. For more information, see Configure synchronization when triggers exist.

• Limits on RENAME TABLE operations

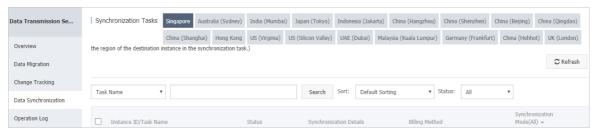
RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

Preparations

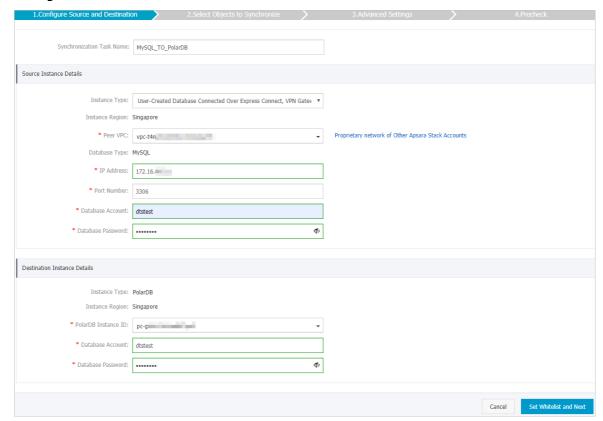
Create an account for a user-created MySQL database and configure binary logging

Note The database accounts must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to MySQL, Target Instance to PolarDB, and Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



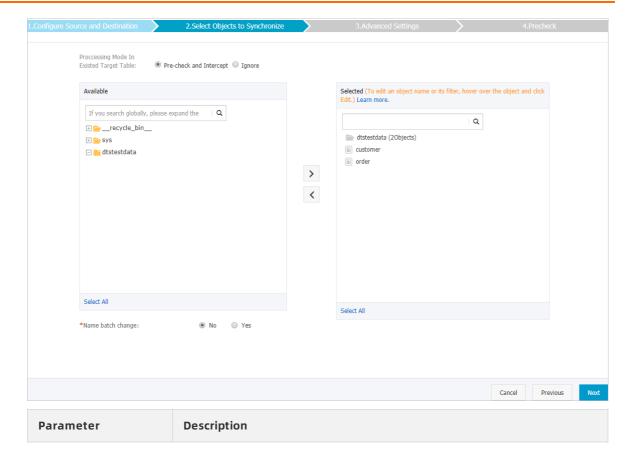
- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



| Section | Parameter | Description |
|---------|----------------------------------|--|
| N/A | Synchroniz ation Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway. |
| | Instance Region | The region of the source instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Peer VPC | Select the ID of the VPC that is connected to the user-created MySQL database. |
| | | |

| §6GiQn | Parameter | Description |
|------------------------------------|------------------------|---|
| Instance Details | Database Type | The value of this parameter is set to MySQL and cannot be changed. |
| | IP Address | Enter the server IP address of the user-created MySQL database. |
| | Port Number | Enter the service port number of the user-created MySQL database. |
| | Database Account | Enter the account of the user-created MySQL database. The account is the same as the database account that you created in Preparations. |
| | Database Password | Enter the password for the account of the user-created MySQL database. |
| Destination Instance Details | Instance Type | The value of this parameter is set to PolarDB and cannot be changed. |
| | Instance Region | The region of the destination instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | PolarDB Instance ID | The ID of the destination PolarDB cluster. |
| | Database Account | Enter the database account of the destination PolarDB cluster. |
| | | Note The database account must have the ALL permission for the objects to be synchronized. |
| | Database Password | Enter the password for the destination database account. |

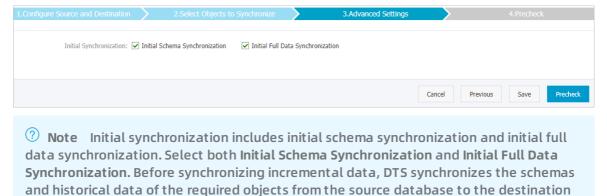
- 7. In the lower-right corner of the page, click Set Whitelist and Next.
 - Note The CIDR blocks of DTS servers are automatically added to the whitelist of the destination PolarDB cluster. This ensures that DTS servers can connect to the destination PolarDB cluster.
- 8. Configure the processing mode in existing destination tables and the objects to be synchronized.



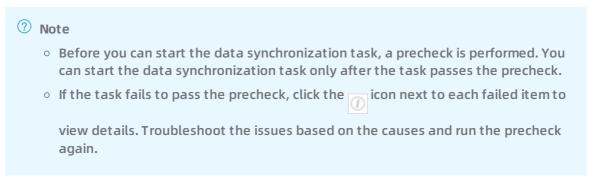
| Parameter | Description | |
|----------------------|--|--|
| | Pre-check and Intercept: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. Note If tables in the destination database have the same | |
| | names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Specify the name of an object in the destination instance. | |
| Processing Mode In | Ignore: skips the precheck for identical table names in the source and destination databases. | |
| Existed Target Table | Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks. DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization. If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. | |
| | Select objects from the Available section and click the jicon to move the objects to the Selected section. | |
| | You can select tables and databases as the objects to be synchronized. | |
| Objects | Note If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination PolarDB cluster by using the object name mapping feature. For more information about how to use this feature, see Specify the name of an object in the destination instance. | |

- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.

database.

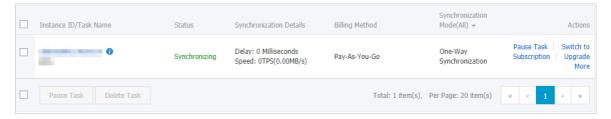


11. In the lower-right corner of the page, click Precheck.



- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the Synchronization Tasks page, view the status of the data synchronization task.



9.Synchronize data to or from a DRDS instance

9.1. Synchronize data between DRDS instances

Distributed Relational Database Service (DRDS) is developed by Alibaba Group to address the bottleneck of single-host database services. DRDS is compatible with the MySQL protocol and syntax, and supports automatic sharding, online smooth scaling, auto scaling, and transparent read/write splitting. DRDS provides O&M capabilities throughout the lifecycle of databases. This topic describes how to synchronize data between DRDS instances by using Data Transmission Service (DTS).

Prerequisites

- The MySQL version of the source and destination DRDS instance is MySQL 5.
- The databases is created in the DRDS instance based on one or more ApsaraDB RDS for MySQL instances that you purchased.
- The tables to be synchronized contain primary keys.
- The destination database in the RDS instance that corresponds to the destination DRDS instance has sufficient storage space.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less
- We recommend that you do not perform operations such as scale-out, migration hot spot table, change shard key, and change DDL during data synchronization. Otherwise, data synchronization may fail.
- After you change the network type of a DRDS instance during data synchronization, you must submit a ticket to modify the network connection information of the synchronization channel.

SQL operations that can be synchronized

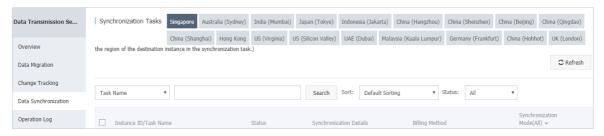
INSERT, UPDATE, and DELETE

Preparations

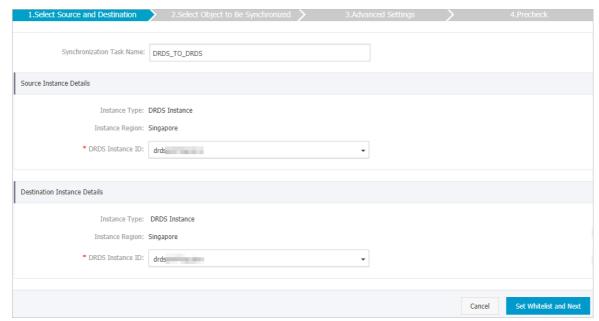
Create a database and tables in the destination DRDS instance based on the schemas of the objects in the source DRDS instance. This is because DTS does not support initial schema synchronization between DRDS instances. For more information, see Create a DRDS database and Create a DRDS table.

? Note Initial schema synchronization: DTS synchronizes the schemas of the required objects from the source database to the destination database.

- 1. Purchase a data synchronization instance.
 - Note On the buy page, set both Source Instance and Target Instance to DRDS and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

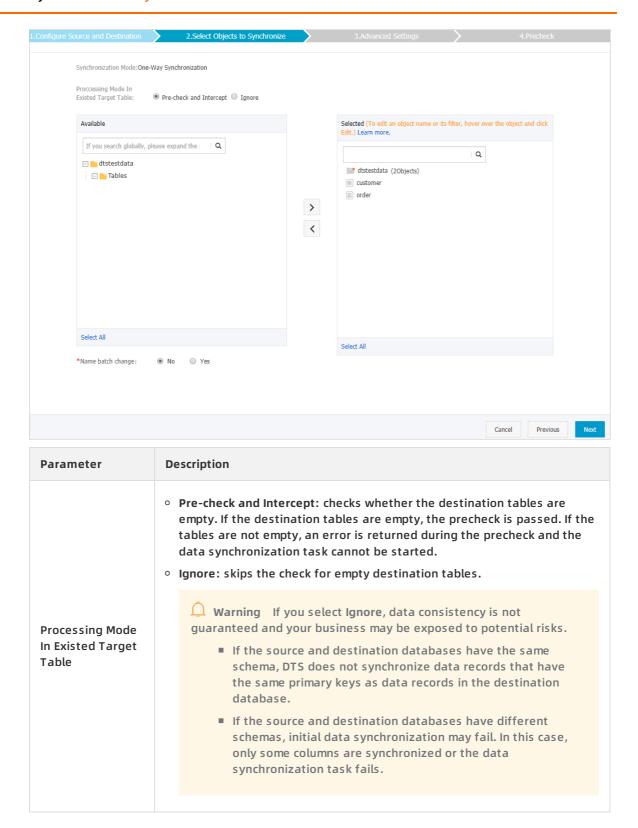


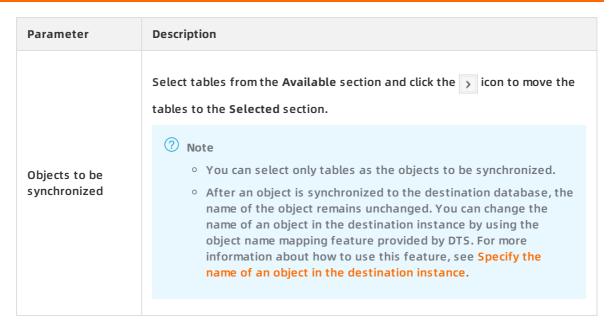
| Section | Parameter | Description |
|------------------------------------|-------------------------------|---|
| N/A | Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | The value of this parameter is set to DRDS and cannot be changed. |
| Source Instance Details | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | DRDS Instance ID | Select the ID of the source DRDS instance. |
| Destination Instance Details | Instance Type | The value of this parameter is set to DRDS and cannot be changed. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | DRDS Instance ID | Select the ID of the destination DRDS instance. |

7. In the lower-right corner of the page, click Set Whitelist and Next.

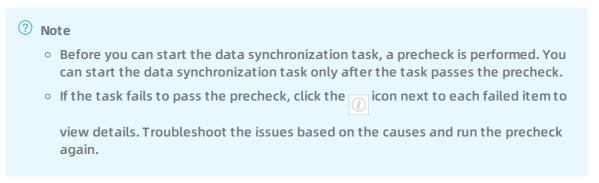
? Note The CIDR blocks of DTS servers are automatically added to the whitelist of the source and destination DRDS instances. This ensures that DTS servers can connect to the DRDS instances.

8. Configure the synchronization policy and objects.



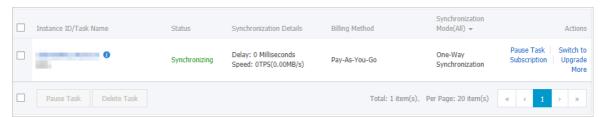


- 9. Click Next.
- 10. Specify whether you want to perform initial full data synchronization.
- 11. In the lower-right corner of the page, click Precheck.



- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the Synchronization Tasks page, view the status of the data synchronization task.



9.2. Synchronize data from a DRDS instance to an AnalyticDB for MySQL cluster

AnalyticDB for MySQL is a real-time online analytical processing (RT-OLAP) service developed by Alibaba Cloud for online data analysis with high concurrency. AnalyticDB for MySQL can analyze petabytes of data from multiple dimensions at millisecond-level timing to provide you with data-driven insights into your business. This topic describes how to synchronize data from a DRDS instance to an AnalyticDB for MySQL cluster by using Data Transmission Service (DTS). AnalyticDB for MySQL allows you to build internal business intelligence (BI) systems, interactive query systems, and real-time report systems.

Prerequisites

- The MySQL version of the DRDS instance is MySQL 5.
- The databases is created in the DRDS instance based on one or more ApsaraDB RDS for MySQL instances that you purchased.
- An AnalyticDB for MySQL cluster is created. For more information, see Create an AnalyticDB for MySQL cluster.
- The destination AnalyticDB for MySQL cluster has sufficient storage space.

Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.
- We recommend that you do not perform operations such as scale-out, migration hot spot table, change shard key, and change DDL during data synchronization. Otherwise, data synchronization may fail.
- After you change the network type of a DRDS instance during data synchronization, you must submit a ticket to modify the network connection information of the synchronization channel.

SQL operations that can be synchronized

Data manipulation language (DML) operations: INSERT, UPDATE, and DELETE

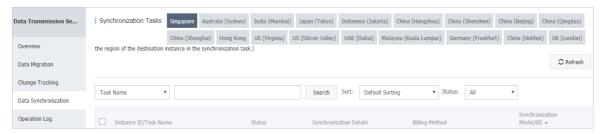
Permissions required for database accounts

| Database | Required permission |
|----------------------|--|
| DRDS | The REPLICATION SLAVE permission, the REPLICATION CLIENT permission, and the permission to perform SELECT operations on the required objects (These permissions are granted by DTS.) |
| AnalyticDB for MySQL | The read/write permissions on the objects to be synchronized |

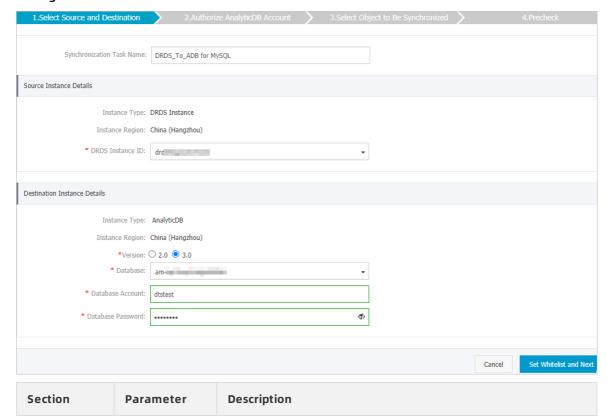
Data type mappings

For more information, see Data type mappings for initial schema synchronization.

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to DRDS, set Target Instance to AnalyticDB for MySQL, and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

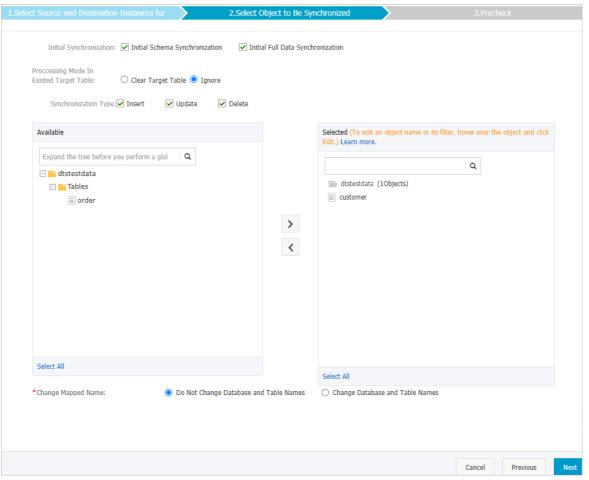


| Section | Parameter | Description | |
|------------------------------------|-------------------------------|---|--|
| N/A | Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. | |
| Source Instance Details | Instance Type | The value of this parameter is set to DRDS and cannot be changed. | |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. | |
| | DRDS Instance ID | Select the ID of the source DRDS instance. | |
| Destination Instance Details | Instance Type | The value of this parameter is set to AnalyticDB and cannot be changed. | |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. | |
| | Version | Select 3.0. | |
| | Database | Select the ID of the destination AnalyticDB for MySQL cluster. | |
| | Database Account | Enter the database account of the AnalyticDB for MySQL cluster. For more information about the permissions that are required for the account, see Permissions required for database accounts. | |
| | Database Password | Enter the password of the destination database account. | |

7. In the lower-right corner of the page, click **Set Whitelist and Next**.

? Note DTS adds the CIDR blocks of DTS servers to the whitelists of the source DRDS instance and the destination AnalyticDB for MySQL cluster. This ensures that DTS servers can connect to the source instance and the destination cluster.

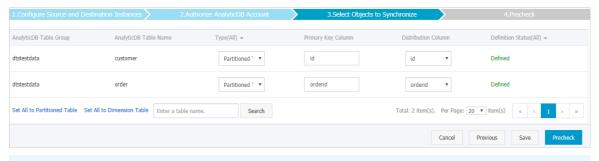
8. Configure the synchronization policy and objects.



| Section | Parameter | Description |
|----------------------------|--|--|
| Synchroniz ation policy | Initial Synchronization | You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization. |
| | Processing Mode In Existed Target Table | Clear Target Table Skips the Schema Name Conflict item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. Ignore |
| | | Skips the Schema Name Conflict item during the precheck. Adds data to the existing data during initial full data synchronization. If you want to synchronize data from multiple tables to one table, you can select this mode. |

| Section | Parameter | Description |
|--------------------------------------|----------------------|---|
| | Synchronization Type | Select the types of operations that you want to synchronize based on your business requirements. Insert Update Delete |
| Objects to be synchroniz ed | N/A | Select tables from the Available section and click the icon to move the tables to the Selected section. |
| | | Note You can select only tables as the objects to be synchronized. You can change the names of the columns that are synchronized to the destination database by using the object name mapping feature. For more information about how to use this feature, see Specify the name of an object in the destination instance. |
| | | |

- 9. In the lower-right corner of the page, click Next.
- 10. Specify a type for the tables that you want to synchronize to the destination database.



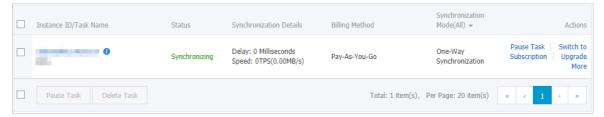
Note After you select Initial Schema Synchronization, you must specify the type, primary key column, and partition key column for the tables to be synchronized to AnalyticDB for MySQL. For more information, see CREATE TABLE.

11. In the lower-right corner of the page, click **Precheck**.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the Synchronization Tasks page, view the status of the data synchronization task.



9.3. Synchronize data from a DRDS instance to an AnalyticDB for PostgreSQL instance

AnalyticDB for PostgreSQL (previously known as HybridDB for PostgreSQL) is a fast, easy-to-use, and cost-effective warehousing service that can process petabytes of data. This topic describes how to synchronize data from a DRDS instance to an AnalyticDB for PostgreSQL instance by using Data Transmission Service (DTS). The data synchronization feature is applicable to scenarios such as ad-hoc query and analysis, extract, transform, and load (ETL) operations, and data visualization.

Prerequisites

- The MySQL version of the DRDS instance is MySQL 5.
- The databases is created in the DRDS instance based on one or more ApsaraDB RDS for MySQL instances that you purchased.
- The tables to be synchronized from the source database contain primary keys.
- An AnalyticDB for PostgreSQL instance is created. For more information, see Create an instance.

Precautions

DTS uses read and write resources of the source and destination databases during initial full
data synchronization. This may increase the database load. If the database performance is
unfavorable, the specification is low, or the data volume is large, database services may
become unavailable. For example, DTS occupies a large amount of read and write resources in
the following cases: a large number of slow SQL queries are performed on the source
database, the tables have no primary keys, or a deadlock occurs in the destination database.

Before synchronizing data, you must evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. For example, you can synchronize data when the CPU usage of the source and destination databases is less than 30%.

- We recommend that you do not perform operations such as scale-out, migration hot spot table, change shard key, and change DDL during data synchronization. Otherwise, data synchronization may fail.
- After you change the network type of a DRDS instance during data synchronization, you must submit a ticket to modify the network connection information of the synchronization channel.
- After a data synchronization task is started, new tables that are created in the source database can also be synchronized. However, you must add the tables to the required objects. For more information see Add an object to a data synchronization task.

Limits

- You can select only tables as the objects to be synchronized.
- DTS does not synchronize the following types of data: JSON, GEOMETRY, CURVE, SURFACE, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, and GEOMETRYCOLLECTION.

SQL operations that can be synchronized

Data manipulation language (DML) operations: INSERT, UPDATE, and DELETE

Term mappings

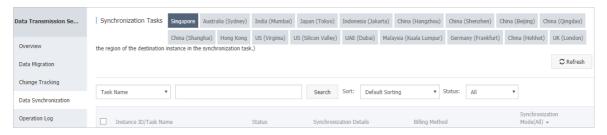
| Term in DRDS | Term in AnalyticDB for PostgreSQL |
|--------------|-----------------------------------|
| Database | Schema |
| Table | Table |

Procedure

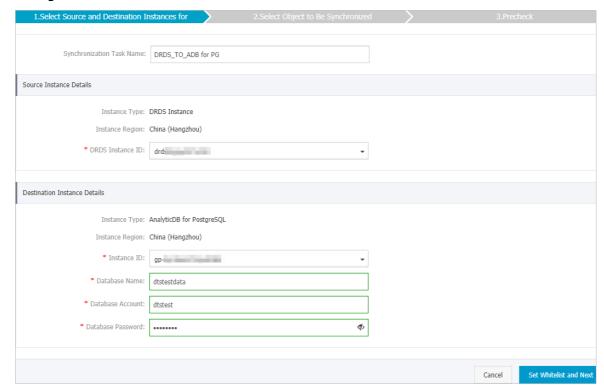
1. Purchase a data synchronization instance.

? Note On the buy page, set Source Instance to DRDS, set Target Instance to AnalyticDB for PostgreSQL, and set Synchronization Topology to One-Way Synchronization.

- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



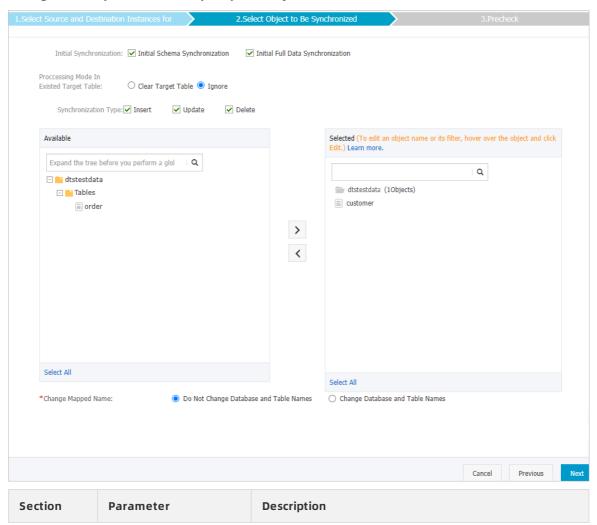
| Section | Parameter | Description |
|-------------------------------|-------------------------------|---|
| N/A | Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | The value of this parameter is set to DRDS and cannot be changed. |
| Source Instance Details | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | DRDS Instance ID | Select the ID of the DRDS instance. |
| | Instance Type | The value of this parameter is set to AnalyticDB for PostgreSQL and cannot be changed. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the AnalyticDB for PostgreSQL instance. |
| Destination Instance | | |

| Petaila | Parameter | Description |
|---------|----------------------|---|
| | Database Name | Enter the name of the destination database in the AnalyticDB for PostgreSQL instance. |
| | Database Account | Enter the database account of the destination AnalyticDB for PostgreSQL instance. |
| | Database Password | Enter the password of the destination database account. |

7. In the lower-right corner of the page, click Set Whitelist and Next.

Note DTS adds the CIDR blocks of DTS servers to the whitelists of the source DRDS instance and the destination AnalyticDB for PostgreSQL instance. This ensures that DTS servers can connect to the source and destination instances.

8. Configure the synchronization policy and objects.



| Section | Parameter | Description |
|--------------------------------------|--|--|
| Synchroniz ation policy | Initial Synchronization | You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization. |
| | Processing Mode In Existed Target Table | Clear Target Table Skips the Schema Name Conflict item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. Ignore Skips the Schema Name Conflict item during the precheck. Adds data to the existing data during initial full data synchronization. If you want to synchronize data from multiple tables to one table, you can select this mode. |
| | Synchronization Type | Select the types of operations that you want to synchronize based on your business requirements. Insert Update Delete |
| Objects to be synchroniz ed | N/A | Select tables from the Available section and click the icon to move the tables to the Selected section. |
| | | Note You can select only tables as the objects to be synchronized. You can change the names of the columns that are synchronized to the destination database by using the object name mapping feature. For more information about how to use this feature, see Specify the name of an object in the destination instance. |

9. In the lower-right corner of the page, click ${\bf Next.}$

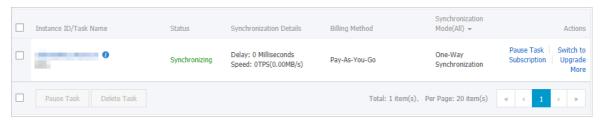
10. Specify the primary key column and distribution column of the table that you want to synchronize to the AnalyticDB for PostgreSQL instance.



? Note The page in this step appears only if you select Initial Schema Synchronization in step 8. For more information about primary key columns and distribution columns, see Table constraints and Partition keys.

- 11. In the lower-right corner of the page, click Precheck.
 - Note
 Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
 If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the Synchronization Tasks page, view the status of the data synchronization task.



10.Synchronize data between PostgreSQL databases

10.1. Synchronize data from an ApsaraDB RDS for PostgreSQL instance to an AnalyticDB for PostgreSQL instance

This topic describes how to synchronize data from an ApsaraDB RDS for PostgreSQL instance to an AnalyticDB for PostgreSQL instance by using Data Transmission Service (DTS). The data synchronization feature provided by DTS allows you to transfer and analyze data with ease.

Prerequisites

- The tables to be synchronized from the ApsaraDB RDS for PostgreSQL instance contain primary kevs.
- An AnalyticDB for PostgreSQL instance is created. For more information, see Create an instance.

Precautions

- A single data synchronization task can synchronize data from only one database. To synchronize data from multiple databases, you must create a data synchronization task for each database.
- During data synchronization, new tables that are created in the source database can also be synchronized. However, to ensure data consistency, you must run the following statement on the new tables before they can be synchronized:

ALTER TABLE schema.table REPLICA IDENTITY FULL;

Limits

- Initial schema synchronization is not supported. DTS does not synchronize the schemas of the required objects from the source database to the destination database.
- You can select only tables as the objects to be synchronized.
- You cannot synchronize the following types of data: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID SNAPSHOT.
- If you perform a DDL operation on an object to be synchronized in the source database during data synchronization, you must perform the operation in the destination database. You must also restart the data synchronization task.

SQL operations that can be synchronized

Only INSERT, UPDATE, and DELETE operations can be synchronized.

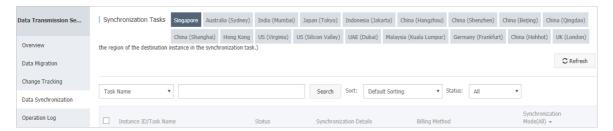
Preparations

1. Change the value of the wal_level parameter for the source RDS instance.

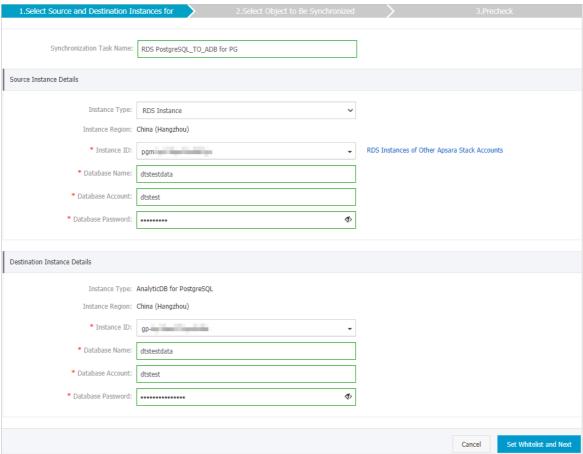
☐ Warning After you change the value of the wal_level parameter, you must restart the instance to apply the modification. We recommend that you evaluate the impact on your business and change the setting during off-peak hours.

- i. Log on to the ApsaraDB for RDS console.
- ii. In the upper-left corner of the page, select the region where the RDS instance resides.
- iii. Find the target RDS instance and click its ID.
- iv. In the left-side navigation pane, click Parameters.
- v. On the **Parameters** page, find the wal_level parameter and change the parameter value to logical .
- Create a database, schema, and table in the destination AnalyticDB for PostgreSQL instance based on the schema of the objects to be synchronized. For more information, see SQL reference.

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to PostgreSQL, set Target Instance to AnalyticDB for PostgreSQL, and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination databases.



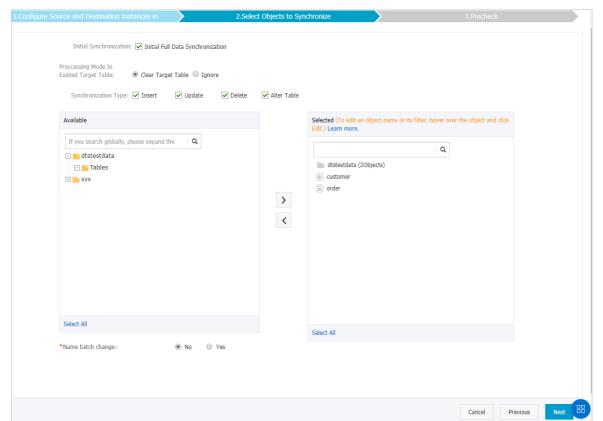
| Section | Parameter | Description |
|----------------------------|------------------------------|---|
| N/A | Synchronization Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select RDS Instance. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the ApsaraDB RDS for PostgreSQL instance. |
| | Database Name | Enter the name of the source database. |
| Source Instance Details | Database Account | Enter the database account of the ApsaraDB RDS for PostgreSQL instance. |
| | | ? Note The account must have the owner permission on schemas. |
| | | |

| Section | Parameter | Description |
|---------------------------------|----------------------|---|
| | Database Password | Enter the password of the source database account. |
| | Instance Type | The value of this parameter is set to AnalyticDB for PostgreSQL and cannot be changed. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the AnalyticDB for PostgreSQL instance. |
| Destination Instance Details | Database Name | Note The destination database must exist in the ApsaraDB RDS for PostgreSQL instance. For more information, see Create a database. |
| | Database Account | Enter the database account of the AnalyticDB for PostgreSQL instance. Note The database account must have the SELECT, INSERT, UPDATE, DELETE, COPY, TRUNCATE, and ALTER TABLE permissions. |
| | Database Password | Enter the password of the destination database account. |

7. In the lower-right corner of the page, click Set Whitelist and Next.

? Note DTS adds the CIDR blocks of DTS servers to the whitelists of the source ApsaraDB RDS for PostgreSQL instance and the destination AnalyticDB for PostgreSQL instance. This ensures that DTS servers can connect to the source and destination instances.

8. Configure the synchronization policy and objects.



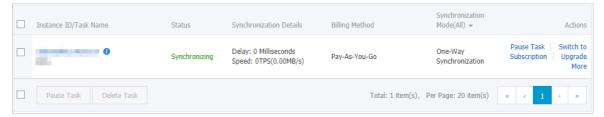
| Section | Parameter | Description |
|----------------------------|--|--|
| | Initial Synchronization | Initial Full Data Synchronization is selected by default. After the precheck, DTS synchronizes historical data of the required objects from the source instance to the destination instance. The data is the basis for subsequent incremental synchronization. |
| | Processing Mode In Existed Target Table | Clear Target Table Skips the Schema Name Conflict item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. Ignore Skips the Schema Name Conflict item during the precheck. Adds data to the existing data during initial full data synchronization. If you want to synchronize data from multiple tables to one table, you can select this mode. |
| Synchroniz ation policy | | |

| Section | Parameter | Description |
|--------------------------------------|----------------------|---|
| | | Select the types of operations that you want to synchronize based on your business requirements. Once The Alter Table operation is not supported. |
| | Synchronization Type | Insert Update Delete Alter Table |
| Objects to be synchroniz ed | N/A | Select tables from the Available section and click the icon to move the tables to the Selected section. Note You can select only tables as the objects to be synchronized. You can change the names of the columns that are synchronized to the destination database by using the object name mapping feature. For more information about how to use this feature, see Specify the name of an object in the destination instance. |



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 10. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 11. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.

On the Synchronization Tasks page, view the status of the data synchronization task.



10.2. Synchronize data from a user-created PostgreSQL database to an AnalyticDB for PostgreSQL instance

This topic describes how to synchronize data from a user-created PostgreSQL database to an AnalyticDB for PostgreSQL instance by using Data Transmission Service (DTS). The data synchronization feature provided by DTS allows you to transfer and analyze data with ease.

Prerequisites

- The tables to be synchronized from the user-created PostgreSQL database contain primary keys.
- An AnalyticDB for PostgreSQL instance is created. For more information, see Create an instance.

Precautions

- A single data synchronization task can synchronize data from only one database. To synchronize data from multiple databases, you must create a data synchronization task for each database.
- During data synchronization, new tables that are created in the source database can also be synchronized. However, to ensure data consistency, you must run the following statement on the new tables before they can be synchronized:

ALTER TABLE schema.table REPLICA IDENTITY FULL;

Limits

- Initial schema synchronization is not supported. DTS does not synchronize the schemas of the required objects from the source database to the destination database.
- You can select only tables as the objects to be synchronized.
- You cannot synchronize the following types of data: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID_SNAPSHOT.
- If you perform a DDL operation on an object to be synchronized in the source database during data synchronization, you must perform the operation in the destination database. You must also restart the data synchronization task.

SQL operations that can be synchronized

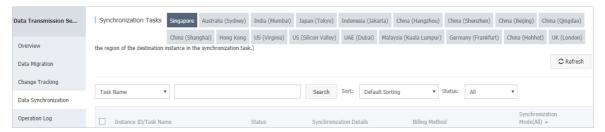
Only INSERT, UPDATE, and DELETE operations can be synchronized.

Preparations

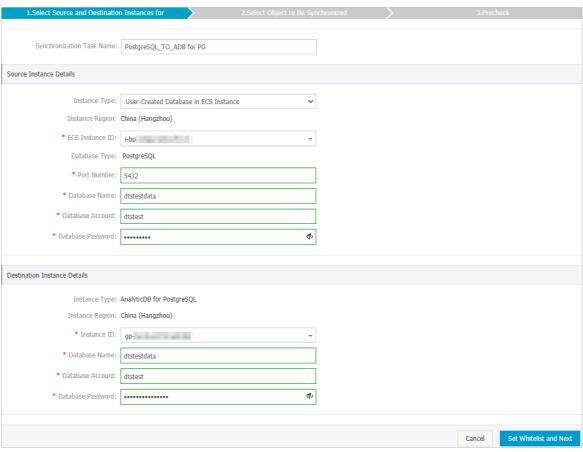
Create a database, schema, and table in the destination AnalyticDB for PostgreSQL instance based on the schema of the objects to be synchronized. For more information, see SQL reference.

Procedure

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to PostgreSQL, set Target Instance to AnalyticDB for PostgreSQL, and set Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination databases.



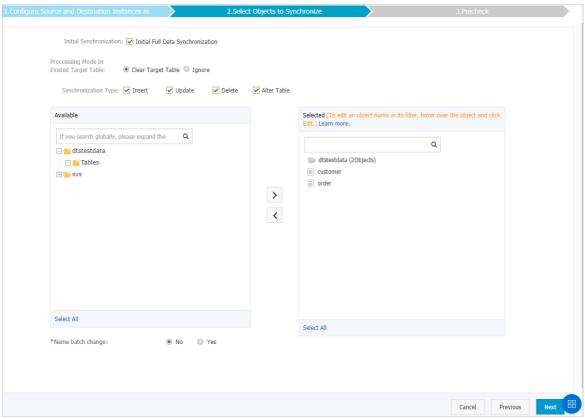
| Section | Parameter | Description |
|----------|------------------------------|---|
| N/A | Synchronization Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select an instance type based on where the source database is deployed. The procedure in this topic uses User-Created Database in ECS Instance as an example. |
| | | Note If you select other instance types, you must prepare the environments that are required for the source database. For more information, see Preparation overview. |
| Instance | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | ECS Instance ID | Select the ID of the ECS instance that is connected to the user-created PostgreSQL database. |

| Source Instance Section Details | Parameter | Description |
|---------------------------------------|----------------------|---|
| | | |
| | Database Type | The value of this parameter is set to PostgreSQL and cannot be changed. |
| | Port Number | Enter the service port number of the user-created PostgreSQL database. |
| | Database Name | Enter the name of the source database. |
| | | Enter the account of the user-created PostgreSQL database. |
| | Database Account | ? Note The account must have the owner permission on schemas. |
| | Database Password | Enter the password of the source database account. |
| | Instance Type | The value of this parameter is set to AnalyticDB for PostgreSQL and cannot be changed. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the AnalyticDB for PostgreSQL instance. |
| Destination | Database Name | Enter the name of the destination database. |
| Instance Details | | Enter the database account of the AnalyticDB for PostgreSQL instance. |
| | Database Account | Note The database account must have the SELECT, INSERT, UPDATE, DELETE, COPY, TRUNCATE, and ALTER TABLE permissions. |
| | Database Password | Enter the password of the destination database account. |

7. In the lower-right corner of the page, click Set Whitelist and Next.

Note DTS adds the CIDR blocks of DTS servers to the inbound rule of the source ECS instance and the whitelist of the destination AnalyticDB for PostgreSQL instance. This ensures that DTS servers can connect to the source and destination instances.

8. Configure the synchronization policy and objects.



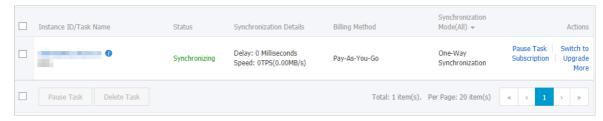
| Section | Parameter | Description |
|------------|--|--|
| | Initial Synchronization | Initial Full Data Synchronization is selected by default. After the precheck, DTS synchronizes historical data of the required objects from the source instance to the destination instance. The data is the basis for subsequent incremental synchronization. |
| Synchroniz | Processing Mode In Existed Target Table | Clear Target Table Skips the Schema Name Conflict item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode. Ignore Skips the Schema Name Conflict item during the precheck. Adds data to the existing data during initial full data synchronization. If you want to synchronize data from multiple tables to one table, you can select this mode. |

| Section | Parameter | Description |
|--------------------------------------|----------------------|---|
| | | Select the types of operations that you want to synchronize based on your business requirements. |
| | | Note The Alter Table operation is not supported. |
| | Synchronization Type | Insert Update Delete Alter Table |
| | e N/A ynchroniz | Select tables from the Available section and click the icon to move the tables to the Selected section. |
| Objects to be synchroniz ed | | Note You can select only tables as the objects to be synchronized. You can change the names of the columns that are synchronized to the destination database by using the object name mapping feature. For more information about how to use this feature, see Specify the name of an object in the destination instance. |
| | | |



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 10. Close the **Precheck** dialog box after the following message is displayed: **The precheck is** passed.
- 11. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

On the Synchronization Tasks page, view the status of the data synchronization task.



11.Synchronize data between Redis databases

11.1. Configure one-way data synchronization between ApsaraDB for Redis instances

Data Transmission Service (DTS) supports one-way data synchronization between ApsaraDB for Redis instances. This feature is applicable to scenarios such as active geo-redundancy and geo-disaster recovery. This topic describes how to configure one-way data synchronization between ApsaraDB for Redis instances.

? Note After you configure a data synchronization task, do not change the architecture type of the source and destination databases. For example, if you change the master-replica architecture to the cluster architecture, data synchronization fails. For more information, see the Architectures section in Overview.

Prerequisites

- The database version of the source ApsaraDB for Redis instance is 4.0 or 5.0.
 - Note The database version of the destination ApsaraDB for Redis instance can be 4.0 or 5.0. The version of the destination database must be the same as or later than the version of the source database. If you want to synchronize data between different versions of Redis databases, make sure that the versions of the source and destination databases are compatible. You can create a pay-as-you-go ApsaraDB for Redis instance to verify database compatibility. Then, you can release the instance or change the billing method to subscription.
- The available storage space of the destination ApsaraDB for Redis instance is larger than the total size of the data in the source ApsaraDB for Redis instance.

Precautions

- DTS uses resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. If you synchronize a large volume of data or the server specifications cannot meet your requirements, database services may become unavailable. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- If the database version of the source ApsaraDB for Redis instance is 2.8, incremental data synchronization is not supported.
- We recommend that you do not run the FLUSHDB or FLUSHALL command in the source instance during data synchronization. Otherwise, data may become inconsistent between the source and destination instances.

If the data eviction policy (maxmemory-policy) of the destination instance is not set to noevic tion , data may become inconsistent between the source and destination instances. For more information about the data eviction policy, see How does ApsaraDB for Redis evict data by default?

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization

For more information about synchronization topologies, see Synchronization topologies.

Operations that can be synchronized

- APPEND
- BITOP, BLPOP, BRPOP, and BRPOPLPUSH
- DECR, DECRBY, and DEL
- EVAL, EVALSHA, EXEC, EXPIRE, and EXPIREAT
- GEOADD and GETSET
- HDEL, HINCRBY, HINCRBYFLOAT, HMSET, HSET, and HSETNX
- INCR, INCRBY, and INCRBYFLOAT
- LINSERT, LPOP, LPUSH, LPUSHX, LREM, LSET, and LTRIM
- MOVE, MSET, MSETNX, and MULTI
- PERSIST, PEXPIRE, PEXPIREAT, PFADD, PFMERGE, PSETEX, and PUBLISH
- RENAME, RENAME, RESTORE, RPOP, RPOPLPUSH, RPUSH, and RPUSHX
- SADD, SDIFFSTORE, SELECT, SET, SETBIT, SETEX, SETNX, SETRANGE, SINTERSTORE, SMOVE, SPOP, SREM, and SUNIONSTORE
- ZADD, ZINCRBY, ZINTERSTORE, ZREM, ZREMRANGEBYLEX, ZUNIONSTORE, ZREMRANGEBYRANK, and ZREMRANGEBYSCORE
- SWAPDB and UNLINK (supported only when the database version of the source ApsaraDB for Redis instance is 4.0)

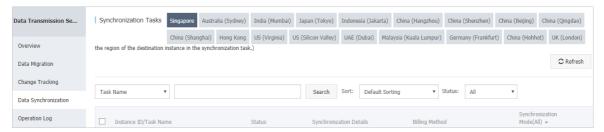
? Note

- If you run the EVAL or EVALSHA command to call Lua scripts, DTS cannot identify
 whether these Lua scripts are executed on the destination database. During
 incremental data synchronization, the destination database does not explicitly return
 the execution results of Lua scripts.
- When DTS calls the SYNC or PSYNC command to transfer data of the LIST type, DTS does not clear the existing data. In this case, the destination database may contain duplicate data records.

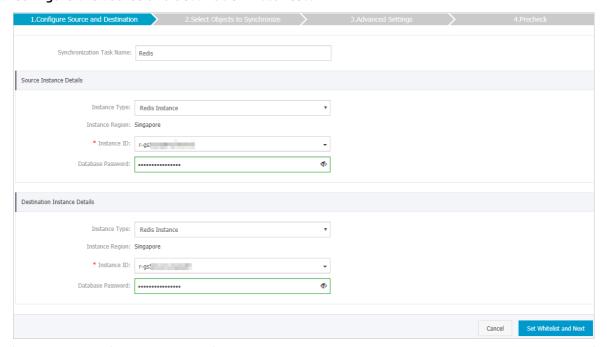
Procedure

1. Purchase a data synchronization instance. For more information, see Purchase a data synchronization instance.

- Note On the buy page, set both Source Instance and Target Instance to Redis.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



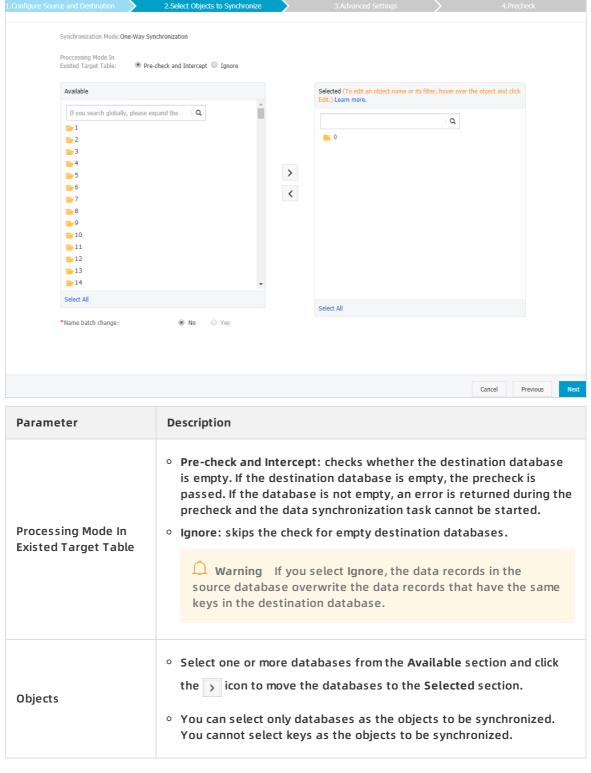
| Section | Parameter | Description |
|---------|-------------------------------|---|
| N/A | Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select Redis Instance. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |

| Section Source | Parameter | Description |
|------------------------------------|----------------------|--|
| Instance Details | Instance ID | Select the ID of the source ApsaraDB for Redis instance. |
| | | Enter the database password of the source ApsaraDB for Redis instance. |
| | Database Password | Note The format of the database password is <user>:<password>. For example, if the username of the custom account is admin and the password is Rp829dlwa, the database password is admin:Rp829dlwa.</password></user> |
| | Instance Type | Select Redis Instance. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the destination ApsaraDB for Redis instance. |
| Destination Instance Details | Database Password | Enter the database password of the destination ApsaraDB for Redis instance. |
| | | Note The format of the database password is <user>:<password>. For example, if the username of the custom account is admin and the password is Rp829dlwa, the database password is admin:Rp829dlwa.</password></user> |
| | | |

7. In the lower-right corner of the page, click Set Whitelist and Next.

Note DTS adds the CIDR blocks of DTS servers to the whitelists of the source and destination ApsaraDB for Redis instances. This ensures that DTS servers can connect to the source and destination ApsaraDB for Redis instances.

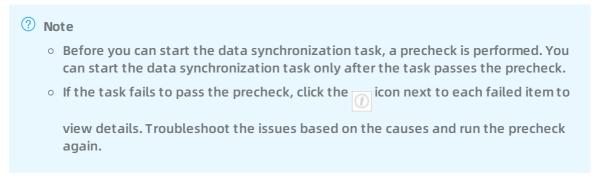
8. Configure the processing mode in existing destination tables and the objects to be synchronized.



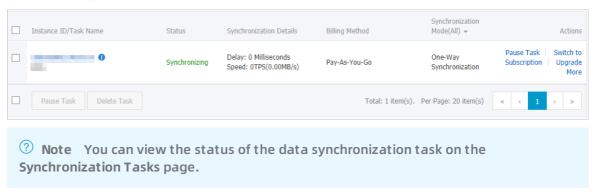
- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization. The value is set to Include full data + incremental data.

Note DTS synchronizes both historical data and incremental data from the source ApsaraDB for Redis instance to the destination ApsaraDB for Redis instance.





- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.



11.2. Configure two-way data synchronization between ApsaraDB for Redis Enhanced Edition instances

This topic describes how to configure two-way data synchronization between ApsaraDB for Redis Enhanced Edition instances by using Data Transmission Service (DTS). The data synchronization feature is applicable to scenarios such as active geo-redundancy and geo-disaster recovery.

Prerequisites

The source and destination instances are ApsaraDB for Redis Enhanced Edition instances (version 5.0). For more information, see the ApsaraDB for Redis Enhanced Edition documentation.

Note ApsaraDB for Redis Enhanced Edition supports the cluster, standard, and read/write splitting architectures.

Precautions

• During two-way data synchronization, the data synchronization task in one direction performs and The data synchronization task in the opposite direction performs only incremental data synchronization.

☐ Warning To ensure data consistency, do not modify or write data to the same key in the source and destination databases while the two-way data synchronization task is running.

- DTS uses resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. If the data volume is large or the specification is low, database services may become unavailable. Before you synchronize data, evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- We recommend that you do not run the FLUSHDB or FLUSHALL command in the source instance. Otherwise, data may be inconsistent between the source and destination instances.
- If the data eviction policy (maxmemory-policy) of the destination database is not set to noevi ction , the data between the source and destination databases may become inconsistent. For more information about the data eviction policy, see How does ApsaraDB for Redis evict data by default?
- If direct connection is disabled for the destination ApsaraDB for Redis instance, DTS uses the proxy forwarding mode to write data to the destination instance.
 - Note For more information, see Enable a direct connection.

Operations that can be synchronized

- APPEND
- BITOP, BLPOP, BRPOP, and BRPOPLPUSH
- DECR, DECRBY, and DEL
- EVAL, EVALSHA, EXEC, EXPIRE, and EXPIREAT
- GEOADD and GETSET
- HDEL, HINCRBY, HINCRBYFLOAT, HMSET, HSET, and HSETNX
- INCR, INCRBY, and INCRBYFLOAT
- LINSERT, LPOP, LPUSH, LPUSHX, LREM, LSET, and LTRIM
- MOVE, MSET, MSETNX, and MULTI
- PERSIST, PEXPIRE, PEXPIREAT, PFADD, PFMERGE, PSETEX, and PUBLISH
- RENAME, RENAMENX, RESTORE, RPOP, RPOPLPUSH, RPUSH, and RPUSHX
- SADD, SDIFFSTORE, SELECT, SET, SETBIT, SETEX, SETNX, SETRANGE, SINTERSTORE, SMOVE, SPOP, SREM, and SUNIONSTORE
- ZADD, ZINCRBY, ZINTERSTORE, ZREM, ZREMRANGEBYLEX, ZUNIONSTORE, ZREMRANGEBYRANK,

and ZREMRANGEBYSCORE

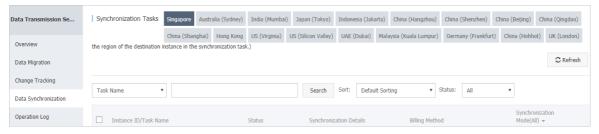
• SWAPDB (This operation is not supported if the source or destination Redis instance is deployed in the cluster architecture.)



- If you use the EVAL or EVALSHA command to call Lua scripts, DTS cannot identify whether these Lua scripts are executed on the destination database. During incremental data synchronization, the destination database does not explicitly return the execution results of Lua scripts.
- When calling the SYNC or PSYNC command to transfer data of the LIST type, DTS does not clear the existing data. In this case, the destination database may contain duplicate data records.

Procedure

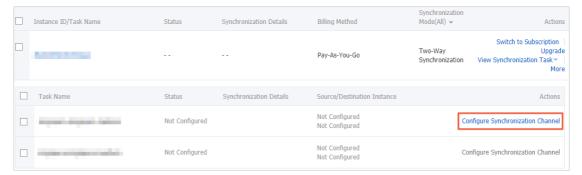
- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Notice On the buy page, set Source Instance to Redis, set Target Instance to Redis, and set Synchronization Topology to Two-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



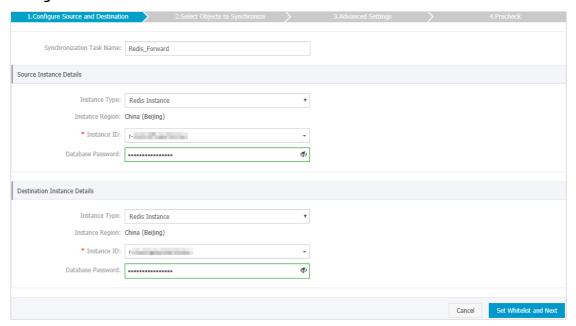
5. Configure the data synchronization task in one direction.

i. Find the data synchronization instance, and click **Configure Synchronization Channel** in the **Actions** column of the first data synchronization task.

Notice A two-way data synchronization instance contains two data synchronization tasks. You must configure a channel for each task. When you configure the second data synchronization task, find the task and click Configure Synchronization Channel in the Actions column.



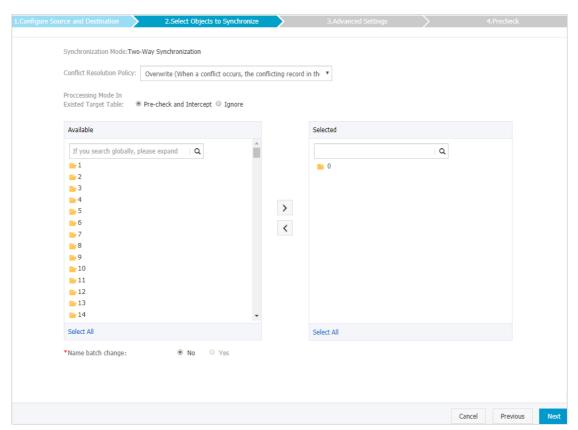
ii. Configure the source and destination instances.



| Section | Parameter | Description |
|---------|----------------------------------|---|
| N/A | Synchronizat ion Task Name | DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select Redis Instance. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |

| Section | Parameter | Description |
|------------------------------------|----------------------|---|
| Source | Instance ID | Select the ID of the source ApsaraDB for Redis instance. |
| Instance Details | | Notice When you configure the data synchronization task in the opposite direction, select the ID of the destination ApsaraDB for Redis instance. |
| | | The database password of the source ApsaraDB for Redis instance. |
| | Database Password | Notice The format of the database password is <user>:<password>. For example, if the username of the custom account is admin and the password is Rp829dlwa, the database password is admin:Rp829dlwa.</password></user> |
| | Instance Type | Select Redis Instance. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the destination ApsaraDB for Redis instance. |
| Destination Instance Details | | Notice When you configure the data synchronization task in the opposite direction, select the ID of the source ApsaraDB for Redis instance. |
| | Database Password | The database password of the destination ApsaraDB for Redis instance. |
| | | Notice The format of the database password is <user>:<pre>cuser>:<pre>cpassword>. For example, if the username of the custom account is admin and the password is Rp829dlwa, the database password is admin:Rp829dlwa.</pre></pre></user> |

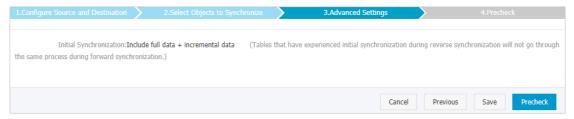
- iii. In the lower-right corner of the page, click Set Whitelist and Next.
 DTS adds the CIDR blocks of DTS servers to the whitelists of the source and destination
 ApsaraDB for Redis instances. This ensures that DTS servers can connect to the source and destination instances.
- iv. Configure the synchronization policy and objects.



| Section | Parameter | Description |
|-------------------------------|---|---|
| Synchro nization policy | Conflict Resolution Policy | Overwrite During data synchronization, if data records have the same key but different values, the data record with the latest key value overwrites the conflicting records. |
| | Processing Mode In Existed Target Table | Pre-check and Intercept: checks whether the destination database is empty. If the destination database is empty, the precheck is passed. If the database is not empty, an error is returned during the precheck and the data synchronization task cannot be started. Ignore: skips the check for empty destination database. Warning If you select Ignore, the data records in the source database overwrite the data records with the same keys in the destination database. |
| | | |

| Section | Parameter | Description |
|--------------------------------------|-----------|--|
| Objects to be synchro nized | N/A | Select a database from the Available section and click the icon to move the database to the Selected section. You can select only databases as the objects to be synchronized. You cannot select keys as the objects to be synchronized. |

- v. In the lower-right corner of the page, click Next.
- vi. Configure initial synchronization.



The value is set to Include full data + incremental data. DTS synchronizes historical data from the source ApsaraDB for Redis instance to the destination ApsaraDB for Redis instance. Then, DTS synchronizes incremental data.

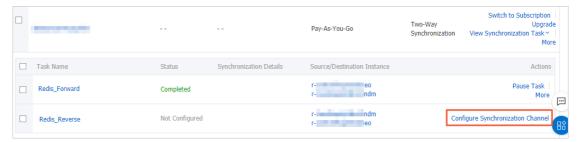
Notice If all the required objects have been synchronized from the source instance to the destination instance, the data synchronization task in the opposite direction synchronizes only incremental data.

vii. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- viii. Close the Precheck dialog box after the following message is displayed: The precheck is passed. Then, the first data synchronization task starts.
- 6. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state. You can view the status of the data synchronization task on the **Synchronization Tasks** page.
- 7. Configure the data synchronization task in the opposite direction.

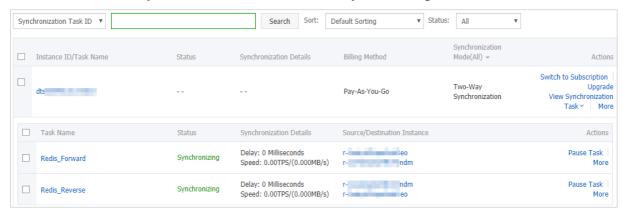
i. Find the second data synchronization task, and click **Configure Synchronization Channel** in the Actions column.



ii. Repeat substeps ii to viii that are described in step 5.

Result

Wait until both data synchronization tasks are in the Synchronizing state.



11.3. Synchronize data from a user-created Redis database hosted on ECS to an ApsaraDB for Redis instance

Data Transmission Service (DTS) supports real-time one-way data synchronization between two Redis databases. This feature is applicable to scenarios such as active geo-redundancy and geo-disaster recovery. This topic describes how to configure one-way data synchronization from a user-created Redis database hosted on ECS to an ApsaraDB for Redis instance.

Note After you configure a data synchronization task, do not change the architecture type of the source and destination databases. For example, if you change the master-replica architecture to the cluster architecture, data synchronization fails. For more information, see the Architectures section in Overview.

Prerequisites

• The version of the source Redis database is 2.8, 3.0, 3.2, 4.0, or 5.0.

Note The database version of the destination ApsaraDB for Redis instance can be 4.0 or 5.0. The version of the destination database must be the same as or later than the version of the source database. If you want to synchronize data between different versions of Redis databases, make sure that the versions of the source and destination databases are compatible. You can create a pay-as-you-go ApsaraDB for Redis instance to verify database compatibility. Then, you can release the instance or change the billing method to subscription.

- The available storage space of the destination ApsaraDB for Redis instance is larger than the total size of the data in the source Redis database.
- All nodes of the Redis cluster support the PSYNC command and share the same password. This is required if the source Redis database is deployed in a cluster architecture.

Precautions

- DTS uses resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. If you synchronize a large volume of data or the server specifications cannot meet your requirements, database services may become unavailable. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- If the bind parameter is configured in the redis.conf file of the source Redis database, set the value of this parameter to the intranet IP address of ECS to ensure that DTS can connect to the source database normally.
- We recommend that you increase the value of the repl-backlog-size parameter in the redis.c onf file of the source Redis database. This ensures the stability of data synchronization.
- To ensure the synchronization quality, DTS adds the following key to the source Redis database: DTS_REDIS_TIMESTAMP_HEARTBEAT . This key is used to record the time when data is synchronized to ApsaraDB for Redis.
- If you configure data synchronization between ApsaraDB for Redis clusters, you cannot run the FLUSHDB or FLUSHALL command in the source cluster. Otherwise, data in the source and destination databases may be inconsistent.
- If the data eviction policy (maxmemory-policy) of the destination database is not set to noevi ction , the data between the source and destination databases may become inconsistent. For more information about the data eviction policy, see How does ApsaraDB for Redis evict data by default?

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization

For more information about synchronization topologies, see Synchronization topologies.

Operations that can be synchronized

APPEND

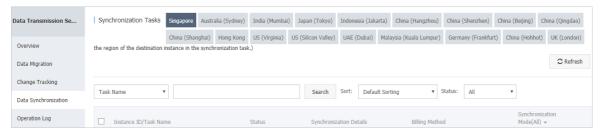
- BITOP, BLPOP, BRPOP, and BRPOPLPUSH
- DECR, DECRBY, and DEL
- EVAL, EVALSHA, EXEC, EXPIRE, and EXPIREAT
- GEOADD and GETSET
- HDEL, HINCRBY, HINCRBYFLOAT, HMSET, HSET, and HSETNX
- INCR, INCRBY, and INCRBYFLOAT
- LINSERT, LPOP, LPUSH, LPUSHX, LREM, LSET, and LTRIM
- MOVE, MSET, MSETNX, and MULTI
- PERSIST, PEXPIRE, PEXPIREAT, PFADD, PFMERGE, PSETEX, and PUBLISH
- RENAME, RENAME, RESTORE, RPOP, RPOPLPUSH, RPUSH, and RPUSHX
- SADD, SDIFFSTORE, SELECT, SET, SETBIT, SETEX, SETNX, SETRANGE, SINTERSTORE, SMOVE, SPOP, SREM, and SUNIONSTORE
- ZADD, ZINCRBY, ZINTERSTORE, ZREM, ZREMRANGEBYLEX, ZUNIONSTORE, ZREMRANGEBYRANK, and ZREMRANGEBYSCORE
- SWAPDB and UNLINK (supported only when the database version of the source ApsaraDB for Redis instance is 4.0)



- If you run the EVAL or EVALSHA command to call Lua scripts, DTS cannot identify
 whether these Lua scripts are executed on the destination database. During
 incremental data synchronization, the destination database does not explicitly return
 the execution results of Lua scripts.
- When DTS calls the SYNC or PSYNC command to transfer data of the LIST type, DTS
 does not clear the existing data. In this case, the destination database may contain
 duplicate data records.

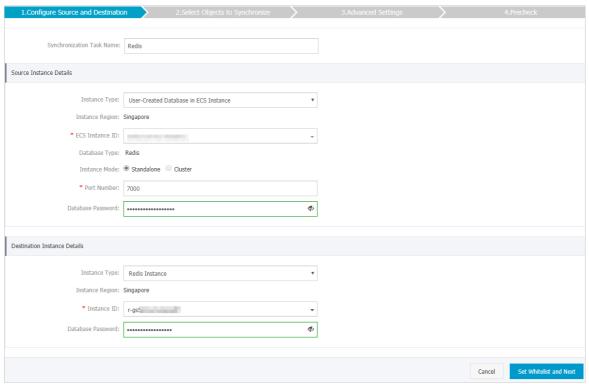
Procedure

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set both Source Instance and Target Instance to Redis.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.

6. Configure the source and destination instances.



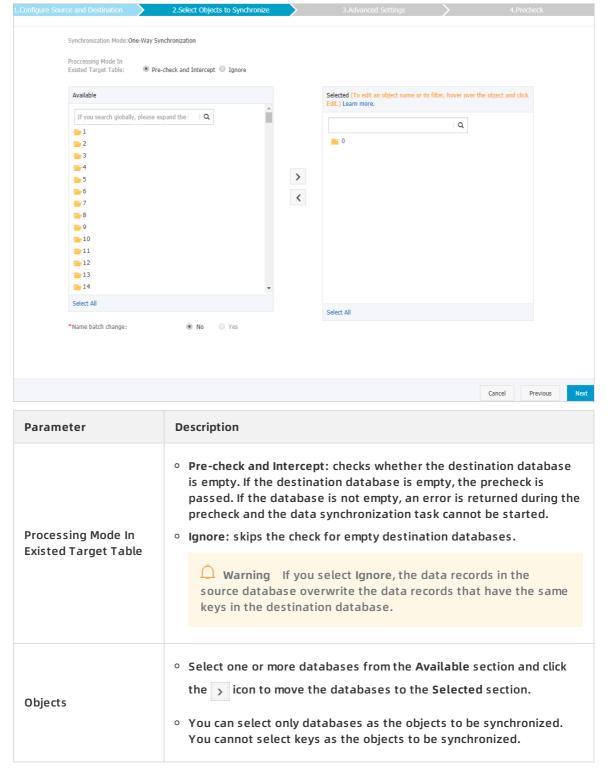
| Section | Parameter | Description |
|--------------------|-------------------------------|---|
| N/A | Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select User-Created Database in ECS Instance. |
| Source Instance | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | ECS Instance ID | Select the ID of the ECS instance that is connected to the source Redis database. |
| | | Note If the source Redis database is deployed in a cluster architecture, select the ID of the ECS instance where a master node resides. |
| | Database Type | The value of this parameter is set to Redis. |
| | Instance Mode | Select Standalone or Cluster based on the architecture of the source Redis database. |
| Details | | |

| Section | Parameter | Description |
|------------------------------------|----------------------|--|
| | Port Number | Enter the service port number of the source Redis database. The default port number is 6379. In this example, 7000 is specified. |
| | | Note If the source Redis database is deployed in a cluster architecture, enter the service port number of a master node. |
| | Database Password | Enter the password for the source Redis database. |
| | | ? Note This parameter is optional and can be left blank if no database password is set. |
| | Instance Type | Select Redis Instance. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the destination ApsaraDB for Redis instance. |
| Destination Instance Details | Database Password | Enter the database password of the destination ApsaraDB for Redis instance. |
| | | Note The format of the database password is <user>:<password>. For example, if the username of the custom account is admin and the password is Rp829dlwa, the database password is admin:Rp829dlwa.</password></user> |
| | | |

7. In the lower-right corner of the page, click Set Whitelist and Next.

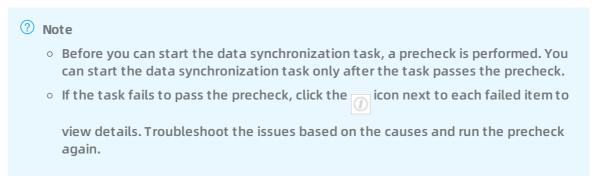
Note The CIDR blocks of DTS servers are automatically added to the inbound rule of the source ECS instance and the whitelist of the destination ApsaraDB for Redis instance. This ensures that DTS servers can connect to the source and destination instances.

8. Configure the processing mode in existing destination tables and the objects to be synchronized.

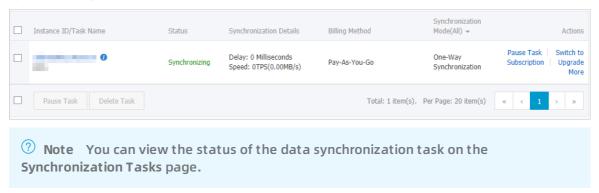


- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization. The value is set to Include full data + incremental data.
 - Note DTS synchronizes both historical data and incremental data from the source ApsaraDB for Redis instance to the destination ApsaraDB for Redis instance.





- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.



11.4. Synchronize data from a user-created Redis database connected over Express Connect, VPN Gateway, or Smart Access Gateway to a user-created Redis database hosted on ECS

Data Transmission Service (DTS) supports real-time one-way data synchronization between two Redis databases. This feature is applicable to scenarios such as active geo-redundancy and geo-disaster recovery. This topic describes how to configure data synchronization from a user-created Redis database connected over Express Connect, VPN Gateway, or Smart Access Gateway to a user-created Redis database hosted on ECS.

Note After you configure a data synchronization task, do not change the architecture type of the source and destination databases. For example, if you change the master-replica architecture to the cluster architecture, data synchronization fails. For more information, see the Architectures section in Overview.

Prerequisites

- The version of the source Redis database is 2.8, 3.0, 3.2, 4.0, or 5.0.
 - ? Note The database version of the destination Redis database can be 2.8, 4.0, or 5.0. The version of the destination database must be the same as or later than the version of the source database. If you want to synchronize data between different versions of Redis databases, make sure that the versions of the source and destination databases are compatible.
- The available storage space of the destination Redis database is larger than the total size of the data in the source Redis database.
- All nodes of the Redis cluster support the PSYNC command and share the same password. This is required if the source Redis database is deployed in a cluster architecture.

Precautions

- DTS uses resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. If you synchronize a large volume of data or the server specifications cannot meet your requirements, database services may become unavailable. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- If the bind parameter is configured in the redis.conf file of the source Redis database, set the value of this parameter to the intranet IP address of ECS to ensure that DTS can connect to the source database normally.
- We recommend that you increase the value of the repl-backlog-size parameter in the redis.c onf file of the source Redis database. This ensures the stability of data synchronization.
- To ensure the synchronization quality, DTS adds the following key to the source Redis database: DTS_REDIS_TIMESTAMP_HEARTBEAT . This key is used to record the time when data is synchronized to ApsaraDB for Redis.
- If you configure data synchronization between ApsaraDB for Redis clusters, you cannot run the FLUSHDB or FLUSHALL command in the source cluster. Otherwise, data in the source and destination databases may be inconsistent.
- If the data eviction policy (maxmemory-policy) of the destination database is not set to noevi ction , the data between the source and destination databases may become inconsistent. For more information about the data eviction policy, see How does ApsaraDB for Redis evict data by default?

Supported synchronization topologies

One-way one-to-one synchronization

- One-way one-to-many synchronization
- One-way cascade synchronization

For more information about synchronization topologies, see Synchronization topologies.

Operations that can be synchronized

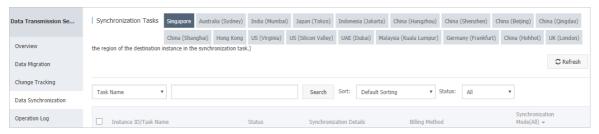
- APPEND
- BITOP, BLPOP, BRPOP, and BRPOPLPUSH
- DECR, DECRBY, and DEL
- EVAL, EVALSHA, EXEC, EXPIRE, and EXPIREAT
- GEOADD and GETSET
- HDEL, HINCRBY, HINCRBYFLOAT, HMSET, HSET, and HSETNX
- INCR, INCRBY, and INCRBYFLOAT
- LINSERT, LPOP, LPUSH, LPUSHX, LREM, LSET, and LTRIM
- MOVE, MSET, MSETNX, and MULTI
- PERSIST, PEXPIRE, PEXPIREAT, PFADD, PFMERGE, PSETEX, and PUBLISH
- RENAME, RENAME, RESTORE, RPOP, RPOPLPUSH, RPUSH, and RPUSHX
- SADD, SDIFFSTORE, SELECT, SET, SETBIT, SETEX, SETNX, SETRANGE, SINTERSTORE, SMOVE, SPOP, SREM, and SUNIONSTORE
- ZADD, ZINCRBY, ZINTERSTORE, ZREM, ZREMRANGEBYLEX, ZUNIONSTORE, ZREMRANGEBYRANK, and ZREMRANGEBYSCORE
- SWAPDB and UNLINK (supported only when the database version of the source ApsaraDB for Redis instance is 4.0)

? Note

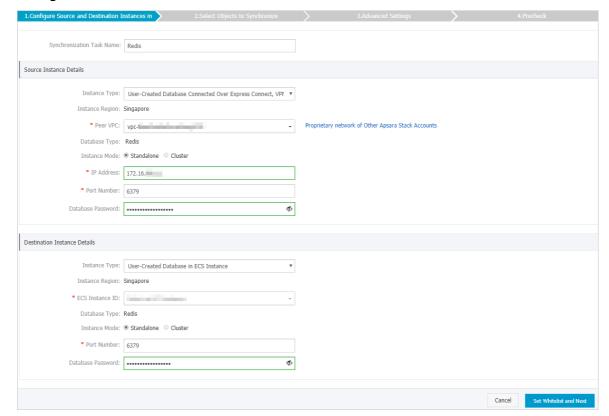
- If you run the EVAL or EVALSHA command to call Lua scripts, DTS cannot identify whether these Lua scripts are executed on the destination database. During incremental data synchronization, the destination database does not explicitly return the execution results of Lua scripts.
- When DTS calls the SYNC or PSYNC command to transfer data of the LIST type, DTS
 does not clear the existing data. In this case, the destination database may contain
 duplicate data records.

Procedure

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set both Source Instance and Target Instance to Redis.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

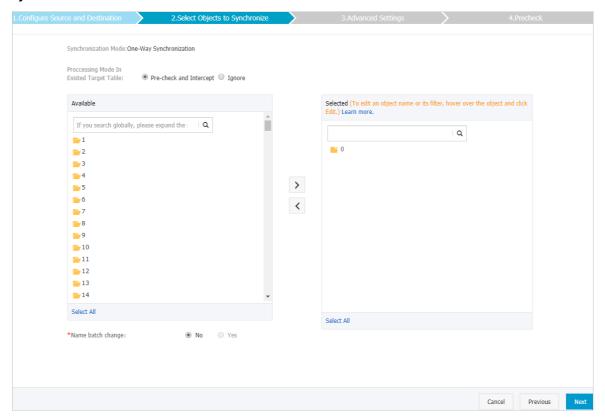


| Section | Parameter | Description |
|---------|-------------------------------|---|
| N/A | Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Peer VPC | Select the ID of the VPC that is connected to the user-created Redis database. |
| | | |

| Section | Parameter | Description |
|------------------------------------|----------------------|---|
| Source Instance Details | Database Type | The value of this parameter is set to Redis. |
| | Instance Mode | Select Standalone or Cluster based on the architecture of the source Redis database. |
| | IP Address | Pote If the source Redis database is deployed in a cluster architecture, enter the IP address of the server to which a master node belongs. |
| | | Enter the service port number of the user-created Redis database. The default port number is 6379. |
| | Port Number | Note If the source Redis database is deployed in a cluster architecture, enter the service port number of a master node. |
| | Database Password | Enter the password for the source Redis database. Note This parameter is optional and can be left blank if no database password is set. |
| | Instance Type | Select User-Created Database in ECS Instance. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the ECS instance that is connected to the destination Redis database. |
| | | Note If the destination Redis database is deployed in a cluster architecture, select the ID of the ECS instance where a master node resides. |
| | Instance Mode | Select Standalone or Cluster based on the architecture of the destination Redis database. |
| Destination Instance Details | | |

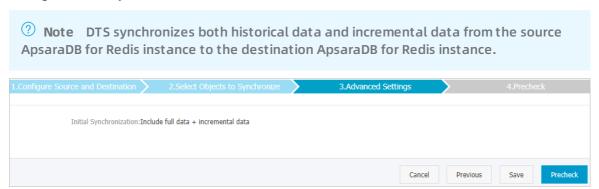
| Section | Parameter | Description |
|---------|----------------------|---|
| | | Enter the service port number of the destination Redis database. The default port number is 6379. |
| | Port Number | Note If the destination Redis database is deployed in a cluster architecture, enter the service port number of a master node. |
| | Database Password | Enter the password for the destination Redis database. |
| | | Enter the password for the destination reals database. |
| | | Note This parameter is optional and can be left blank if no database password is set. |
| | | |
| | | |

- 7. In the lower-right corner of the page, click Set Whitelist and Next.
 - Note The CIDR blocks of DTS servers are automatically added to an inbound rule of the destination ECS instance. This ensures that DTS servers can connect to the destination ECS instance.
- 8. Configure the processing mode in existing destination tables and the objects to be synchronized.



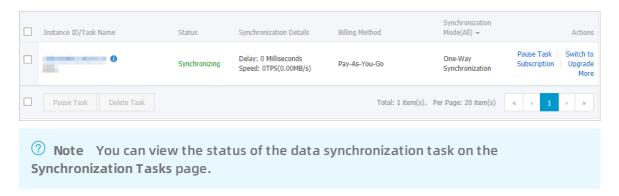
| Parameter | Description | |
|--|--|--|
| Processing Mode In Existed Target Table | Pre-check and Intercept: checks whether the destination database is empty. If the destination database is empty, the precheck is passed. If the database is not empty, an error is returned during the precheck and the data synchronization task cannot be started. Ignore: skips the check for empty destination databases. Warning If you select Ignore, the data records in the source database overwrite the data records that have the same keys in the destination database. | |
| Objects | Select one or more databases from the Available section and click the icon to move the databases to the Selected section. You can select only databases as the objects to be synchronized. You cannot select keys as the objects to be synchronized. | |

- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization. The value is set to Include full data + incremental data.





- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.



11.5. Synchronize data from a user-created Redis cluster to an ApsaraDB for Redis cluster instance

Data Transmission Service (DTS) supports real-time one-way data synchronization between two Redis clusters. This feature is applicable to scenarios such as data migration, active georedundancy, and geo-disaster recovery. This topic describes how to configure one-way data synchronization from a user-created Redis cluster to an ApsaraDB for Redis cluster instance.

You can also follow the procedure to configure data synchronization from an ApsaraDB for Redis cluster instance to a user-created Redis cluster. However, you must configure parameters for the source and destination instances based on the actual scenarios.

Note After you configure the data synchronization task, do not change the architecture type of the source and destination databases. For example, you cannot change the master-replica architecture to the cluster architecture. Otherwise, data synchronization fails. For more information, see the Architectures section in Overview.

Prerequisites

- The version of the user-created Redis database is 2.8, 3.0, 3.2, 4.0, or 5.0.
 - Note The database version of the destination ApsaraDB for Redis cluster instance can be 2.8, 4.0, or 5.0. The version of the destination database must be the same as or later than the version of the source database. If you want to synchronize data between different versions of Redis databases, make sure that the versions of the source and destination databases are compatible. You can create a pay-as-you-go ApsaraDB for Redis cluster instance to verify database compatibility. After verification, you can release the instance or change the billing method to subscription.
- The available storage space of the destination ApsaraDB for Redis cluster instance is larger than the total size of the data in the source Redis database.
- All nodes of the source Redis cluster support the PSYNC command and share the same password.

Precautions

- DTS uses resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. If the data volume is large or the specification is low, database services may become unavailable. Before you synchronize data, evaluate the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- We recommend that you increase the value of the *repl-backlog-size* parameter in the redis.co nf file. This ensures the stability of data synchronization.
- To ensure the synchronization quality, DTS adds the following key to the source Redis database: DTS_REDIS_TIMESTAMP_HEARTBEAT. This key is used to record the time when data is synchronized to ApsaraDB for Redis.
- We recommend that you do not run the FLUSHDB or FLUSHALL command in the source Redis cluster. Otherwise, data in the source and destination databases may be inconsistent.
- If the data eviction policy (maxmemory-policy) of the destination database is not set to noeviction, the data between the source and destination databases may become inconsistent. For more information about the data eviction policy, see How does Apsarable for Redis evict data by default?

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization

For more information about synchronization topologies, see Synchronization topologies.

Operations that can be synchronized

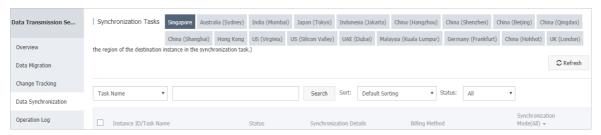
- APPEND
- BITOP, BLPOP, BRPOP, and BRPOPLPUSH
- DECR, DECRBY, and DEL
- EVAL, EVALSHA, EXEC, EXPIRE, and EXPIREAT
- GEOADD and GETSET
- HDEL, HINCRBY, HINCRBYFLOAT, HMSET, HSET, and HSETNX
- INCR, INCRBY, and INCRBYFLOAT
- LINSERT, LPOP, LPUSH, LPUSHX, LREM, LSET, and LTRIM
- MOVE, MSET, MSETNX, and MULTI
- PERSIST, PEXPIRE, PEXPIREAT, PFADD, PFMERGE, PSETEX, and PUBLISH
- RENAME, RENAME, RESTORE, RPOP, RPOPLPUSH, RPUSH, and RPUSHX
- SADD, SDIFFSTORE, SELECT, SET, SETBIT, SETEX, SETNX, SETRANGE, SINTERSTORE, SMOVE, SPOP, SREM, and SUNIONSTORE
- ZADD, ZINCRBY, ZINTERSTORE, ZREM, ZREMRANGEBYLEX, ZUNIONSTORE, ZREMRANGEBYRANK, and ZREMRANGEBYSCORE
- SWAPDB and UNLINK (supported only when the version of the source Redis cluster is 4.0)

? Note

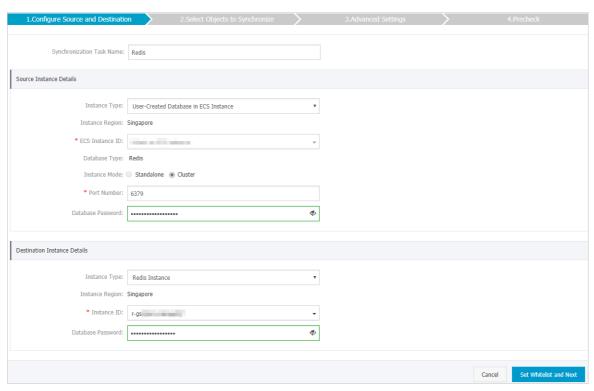
- If you use the EVAL or EVALSHA command to call Lua scripts, DTS cannot identify
 whether these Lua scripts are executed on the destination database. During
 incremental data synchronization, the destination database does not explicitly return
 the execution results of Lua scripts.
- When calling the SYNC or PSYNC command to transfer data of the LIST type, DTS does not clear the existing data. In this case, duplicate data may exist in the destination database.

Procedure

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set both Source Instance and Target Instance to Redis.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



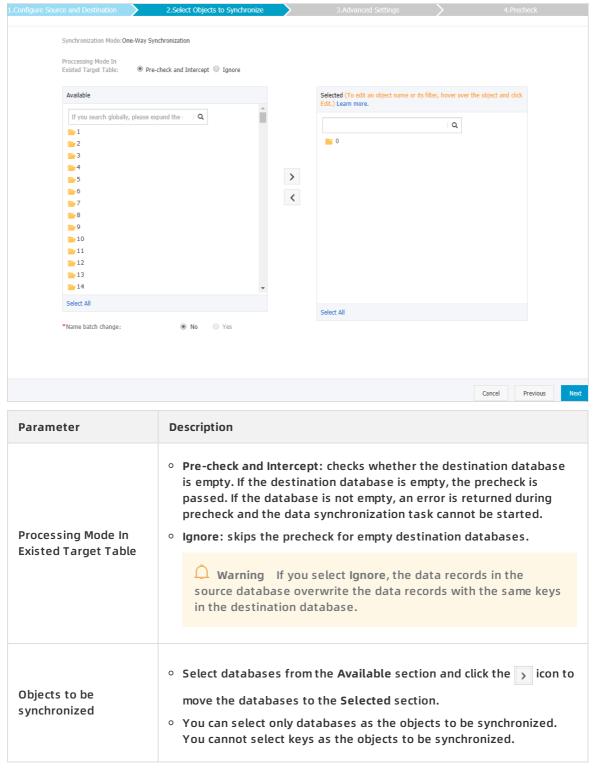
| Section | Parameter | Description |
|-------------------------------|-------------------------------|--|
| N/A | Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| Source Instance Details | Instance Type | Select User-Created Database in ECS Instance. You can select User-Created Database in ECS Instance or User-Created Database Connected Over Express Connect, VPN Gateway, or Smart Access Gateway based on the type of the source database. |
| | | The procedure in this topic uses user-created database hosted on ECS as an example. You can also follow the procedure to configure data synchronization for other types of user-created Redis databases. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | ECS Instance | Select the ID of the ECS instance where a master node in the user-created Redis cluster resides. |
| | Database Type | The value of this parameter is set to Redis. |
| | Instance Mode | Select Cluster. |
| | | |

| Section | Parameter | Description |
|------------------------------------|----------------------|--|
| | Port Number | Enter the service port number of a master node in the user-created Redis cluster. In this example, enter 7000. |
| | Database Password | Enter the password for the user-created Redis database. |
| | | Note This parameter is optional and can be left blank if no database password is set. |
| | Instance Type | Select Redis Instance. |
| Destination Instance Details | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the destination ApsaraDB for Redis cluster instance. |
| | Database Password | Enter the database password of the destination ApsaraDB for Redis cluster instance. |
| | | Note The format of the database password is <user>:<password>. For example, if the username of the custom account is admin and the password is Rp829dlwa, the database password is admin:Rp829dlwa.</password></user> |

7. In the lower-right corner of the page, click Set Whitelist and Next.

Note The CIDR blocks of DTS servers are automatically added to the inbound rule of the source ECS instance and the whitelist of the destination ApsaraDB for Redis cluster instance. This ensures that DTS servers can connect to the source and destination instances.

8. Configure the processing mode in existing destination tables and the objects to be synchronized.

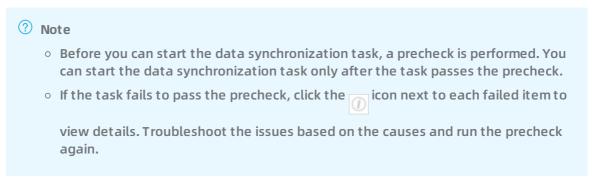


- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization. The value is set to Include full data + incremental data.

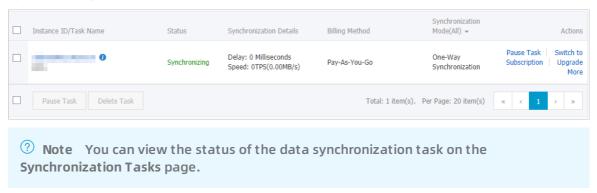
Note Before synchronizing incremental data, DTS synchronizes historical data from the source Redis database to the destination Redis database.



11. In the lower-right corner of the page, click Precheck.



- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.



11.6. Synchronize data from a Codis cluster hosted on ECS to an ApsaraDB for Redis instance

ApsaraDB for Redis is a database service compatible with the open-source Redis protocol and provides hybrid storage of memory and hard disks. Based on reliable hot standby architecture and scalable cluster architecture, ApsaraDB for Redis is suitable for scenarios that require flexible configuration changes, high throughput, and low latency. This topic describes how to synchronize data from a Codis cluster to an ApsaraDB for Redis instance by using Data Transmission Service (DTS).

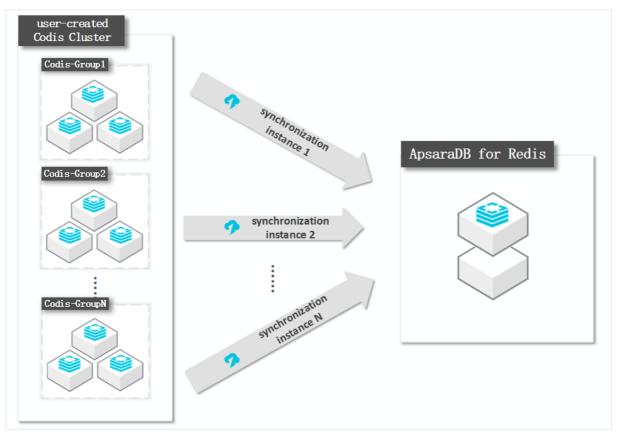
Prerequisites

• The available storage space of the destination ApsaraDB for Redis instance is larger than the total size of data stored in the source Codis cluster.

• All master nodes in the source Codis cluster support the PSYNC command.

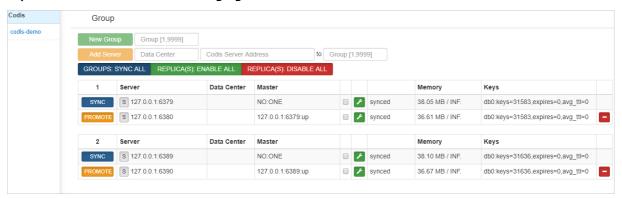
How DTS synchronizes data from a Codis cluster

A Codis cluster consists of multiple Codis-Groups. You must create a data synchronization task for each Codis-Group. DTS synchronizes each Codis-Group in a data synchronization task till the whole cluster is synchronized.



Architecture of the Codis cluster

In this topic, the Codis cluster consists of two Codis-Groups. Each Codis-Group runs in a master-replica architecture. The following figure shows the architecture of the cluster.



Precautions

• DTS uses resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. If you synchronize a large

volume of data or the server specifications cannot meet your requirements, database services may become unavailable. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.

- If the bind parameter is configured in the redis.conf file of the source Redis database, set the value of this parameter to the intranet IP address of ECS to ensure that DTS can connect to the source database normally.
- We recommend that you increase the value of the repl-backlog-size parameter in the redis.c onf file. This ensures the stability of data synchronization.
- To ensure the synchronization quality, DTS adds the following key to the source Codis cluster: DTS_REDIS_TIMESTAMP_HEARTBEAT. This key is used to record the time when data is synchronized to ApsaraDB for Redis.
- We recommend that you do not run the cluster. Otherwise, data may be inconsistent between the Codis cluster and the ApsaraDB for Redis instance.
- If the data eviction policy (maxmemory-policy) of the destination database is not set to noevi ction , the data between the source and destination databases may become inconsistent. For more information about the data eviction policy, see How does ApsaraDB for Redis evict data by default?
- The database version of the destination ApsaraDB for Redis instance can be 2.8, 4.0, and 5.0. The version of the destination database must be the same as or later than the version of the source database. If you want to synchronize data between different versions of Redis databases, make sure that the versions of the source and destination databases are compatible. You can create a pay-as-you-go ApsaraDB for Redis instance to verify database compatibility. After verification, you can release the instance or change the billing method to subscription.

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization

For more information about synchronization topologies, see Synchronization topologies.

Operations that can be synchronized

- APPEND
- BITOP, BLPOP, BRPOP, and BRPOPLPUSH
- DECR, DECRBY, and DEL
- EVAL, EVALSHA, EXEC, EXPIRE, and EXPIREAT
- GEOADD and GETSET
- HDEL, HINCRBY, HINCRBYFLOAT, HMSET, HSET, and HSETNX
- INCR, INCRBY, and INCRBYFLOAT
- LINSERT, LPOP, LPUSH, LPUSHX, LREM, LSET, and LTRIM
- MOVE, MSET, MSETNX, and MULTI

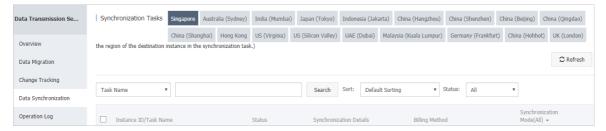
- PERSIST, PEXPIRE, PEXPIREAT, PFADD, PFMERGE, PSETEX, and PUBLISH
- RENAME, RENAME, RESTORE, RPOP, RPOPLPUSH, RPUSH, and RPUSHX
- SADD, SDIFFSTORE, SELECT, SET, SETBIT, SETEX, SETNX, SETRANGE, SINTERSTORE, SMOVE, SPOP, SREM, and SUNIONSTORE
- ZADD, ZINCRBY, ZINTERSTORE, ZREM, ZREMRANGEBYLEX, ZUNIONSTORE, ZREMRANGEBYRANK, and ZREMRANGEBYSCORE



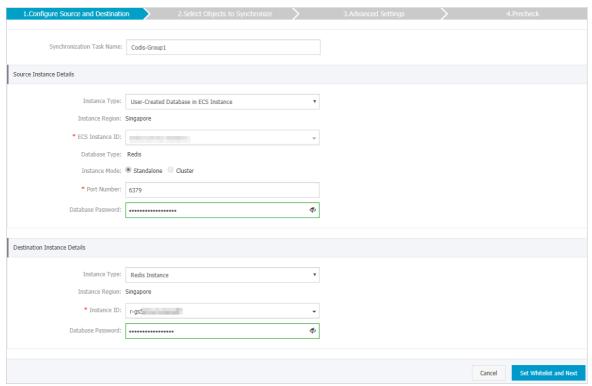
- If you run the EVAL or EVALSHA command to call Lua scripts, DTS cannot identify whether these Lua scripts are executed on the destination database. During incremental data synchronization, the destination database does not explicitly return the execution results of Lua scripts.
- When DTS calls the SYNC or PSYNC command to transfer data of the LIST type, DTS
 does not clear the existing data. In this case, the destination database may contain
 duplicate data records.

Procedure

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - Note On the buy page, set Source Instance to Redis, Target Instance to Redis, and Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click **Data Synchronization**.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.



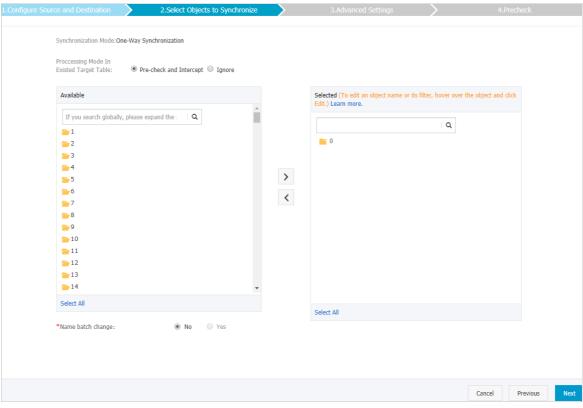
| Section | Parameter | Description |
|--------------------|-------------------------------|--|
| N/A | Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select User-Created Database in ECS Instance. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | ECS Instance ID | Select the ID of the ECS instance where the master node of the Codis-Group resides. |
| | | Note DTS synchronizes each Codis-Group of the Codis cluster by using a data synchronization task till the whole cluster is synchronized. In this step, enter the ECS instance ID for the master node of Codis-Group 1. When you configure the data synchronization task for Codis-Group 2, enter the ECS instance ID for the master node of Codis-Group 2. You can configure data synchronization tasks for all Codis-Groups by following the procedure described in this topic. |
| Source Instance | Database Type | The value of this parameter is set to Redis. |
| Details | | |

| Section | Parameter | Description |
|---------------------|----------------------|--|
| | Instance Mode | Select Standalone. |
| | | ? Note You must select Standalone for this parameter because data from a Codis cluster cannot be synchronized at a time. DTS synchronizes each Codis-Group of the cluster in a data synchronization task till all Codis-Groups are synchronized. |
| | Port Number | Enter the service port number of the master node in the Codis-Group. |
| | | The database password for the master node. |
| | Database Password | Note This parameter is optional and can be left blank if no database password is set. |
| | Instance Type | Select Redis Instance. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| Destination | Instance ID | Select the ID of the destination ApsaraDB for Redis instance. |
| Instance Details | Database Password | Enter the database password of the destination ApsaraDB for Redis instance. |
| | | ? Note The format of the database password is <user>:<password>. For example, if the username of the custom account is admin and the password is Rp829dlwa, the database password is admin:Rp829dlwa.</password></user> |

7. In the lower-right corner of the page, click Set Whitelist and Next.

Note The CIDR blocks of DTS servers are automatically added to the inbound rule of the ECS instance and the whitelist of the ApsaraDB for Redis instance. This ensures that DTS servers can connect to the source and destination instances.

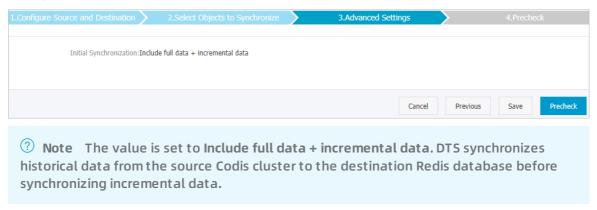
8. Configure the processing mode in existing destination tables and the objects to be synchronized.



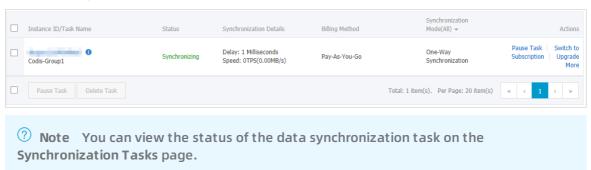
Parameter Description DTS synchronizes each Codis-Group of the Codis cluster in a data synchronization task till the whole cluster is synchronized. When you configure data synchronization for Codis-Group 1, if the ApsaraDB for Redis instance has no data, select Pre-check and Intercept. When you configure data synchronization for Codis-Groups 2 to N, select Ignore. Otherwise, errors may occur during data synchronization. ? Note • Pre-check and Intercept: checks whether the destination Processing Mode In database is empty. If the destination database is empty, the precheck is passed. If the database is not empty, an **Existed Target Table** error is returned during precheck and the data synchronization task cannot be started. o Ignore: skips the precheck for empty destination databases and continues with data synchronization. If the keys in the destination database are the same as those in the source database during data synchronization, the keys in the source database overwrite those in the destination database.

| Parameter | Description |
|----------------------------|--|
| Objects to be synchronized | Select databases from the Available section and click the icon to move the databases to the Selected section. You can select only databases as the objects to be synchronized. You cannot select keys as the objects to be synchronized. |

- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.



- 11. In the lower-right corner of the page, click Precheck.
 - Note
 Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
 If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

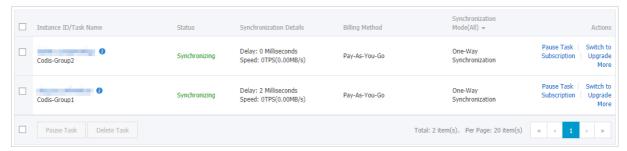


14. Create and configure a data synchronization task for the other Codis-Group by following

steps 1 to 13.

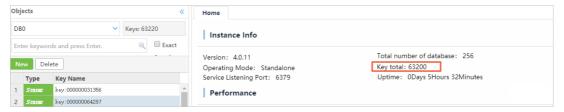
Result

In this topic, the Codis cluster consists of two Codis-Groups. You must create two data synchronization tasks. The following figure shows that the initial synchronization is complete for both tasks and both tasks are in the **Synchronizing** state.

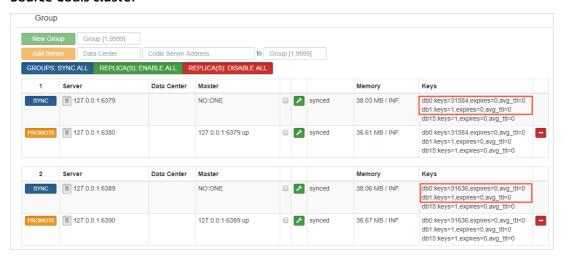


In this topic, the databases DB0 and DB1 are synchronized. You can use Data Management (DMS) to log on to the ApsaraDB for Redis instance and check the total number of keys in the ApsaraDB for Redis instance. The total number of keys is the same as that in the source Codis cluster.

ApsaraDB for Redis instance



Source Codis cluster



11.7. Synchronize data from a Twemproxy Redis cluster hosted on ECS to an ApsaraDB for Redis instance

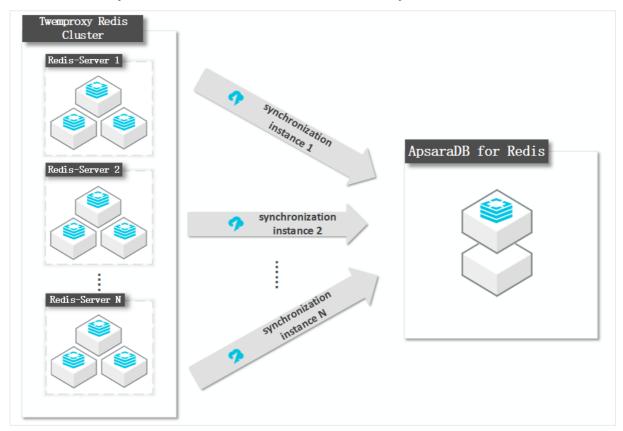
ApsaraDB for Redis is a database service compatible with the open-source Redis protocol and provides hybrid storage of memory and hard disks. Based on reliable hot standby architecture and scalable cluster architecture, ApsaraDB for Redis is suitable for scenarios that require flexible configuration changes, high throughput, and low latency. This topic describes how to synchronize data from a Twemproxy Redis cluster to an ApsaraDB for Redis instance by using Data Transmission Service (DTS).

Prerequisites

- The available storage space of the destination ApsaraDB for Redis instance is larger than the total size of data stored in the source Twemproxy Redis cluster.
- All master nodes in the source Twemproxy Redis cluster support the PSYNC command.

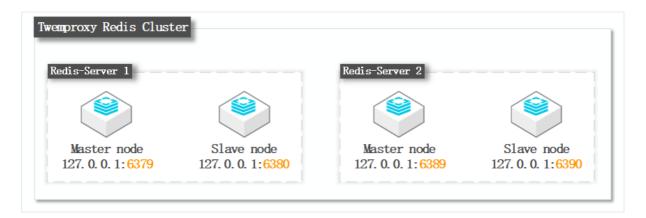
How DTS synchronizes data from a Twemproxy Redis cluster

A Twemproxy Redis cluster consists of multiple Redis-Servers. DTS synchronizes each Redis-Server in a data synchronization task till the whole cluster is synchronized.



Architecture of the Twemproxy Redis cluster

In this topic, the Twemproxy Redis cluster consists of two Redis-Servers. Each Redis-Server runs in a master-replica architecture. The following figure shows the architecture of the cluster.



Precautions

- DTS uses resources of the source and destination databases during initial full data synchronization. This may increase the load of the database server. If you synchronize a large volume of data or the server specifications cannot meet your requirements, database services may become unavailable. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- If the bind parameter is configured in the redis.conf file of the source Redis database, set the value of this parameter to the intranet IP address of ECS to ensure that DTS can connect to the source database normally.
- We recommend that you increase the value of the repl-backlog-size parameter in the redis.c onf file. This ensures the stability of data synchronization.
- To ensure the synchronization quality, DTS adds the following key to the source Codis cluster:

 DTS_REDIS_TIMESTAMP_HEARTBEAT

 . This key is used to record the time when data is synchronized to ApsaraDB for Redis.
- We recommend that you do not run the FLUSHDB or FLUSHALL command in the source Codis cluster. Otherwise, data may be inconsistent between the Codis cluster and the ApsaraDB for Redis instance.
- If the data eviction policy (maxmemory-policy) of the destination database is not set to noevi ction , the data between the source and destination databases may become inconsistent. For more information about the data eviction policy, see How does ApsaraDB for Redis evict data by default?
- The database version of the destination ApsaraDB for Redis instance can be 2.8, 4.0, and 5.0. The version of the destination database must be the same as or later than the version of the source database. If you want to synchronize data between different versions of Redis databases, make sure that the versions of the source and destination databases are compatible. You can create a pay-as-you-go ApsaraDB for Redis instance to verify database compatibility. After verification, you can release the instance or change the billing method to subscription.

Supported synchronization topologies

- One-way one-to-one synchronization
- One-way one-to-many synchronization
- One-way cascade synchronization

For more information about synchronization topologies, see Synchronization topologies.

Operations that can be synchronized

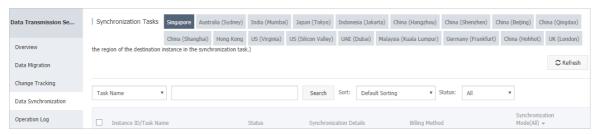
- APPEND
- BITOP, BLPOP, BRPOP, and BRPOPLPUSH
- DECR, DECRBY, and DEL
- EVAL, EVALSHA, EXEC, EXPIRE, and EXPIREAT
- GEOADD and GETSET
- HDEL, HINCRBY, HINCRBYFLOAT, HMSET, HSET, and HSETNX
- INCR, INCRBY, and INCRBYFLOAT
- LINSERT, LPOP, LPUSH, LPUSHX, LREM, LSET, and LTRIM
- MOVE, MSET, MSETNX, and MULTI
- PERSIST, PEXPIRE, PEXPIREAT, PFADD, PFMERGE, PSETEX, and PUBLISH
- RENAME, RENAME, RESTORE, RPOP, RPOPLPUSH, RPUSH, and RPUSHX
- SADD, SDIFFSTORE, SELECT, SET, SETBIT, SETEX, SETNX, SETRANGE, SINTERSTORE, SMOVE, SPOP, SREM, and SUNIONSTORE
- ZADD, ZINCRBY, ZINTERSTORE, ZREM, ZREMRANGEBYLEX, ZUNIONSTORE, ZREMRANGEBYRANK, and ZREMRANGEBYSCORE

? Note

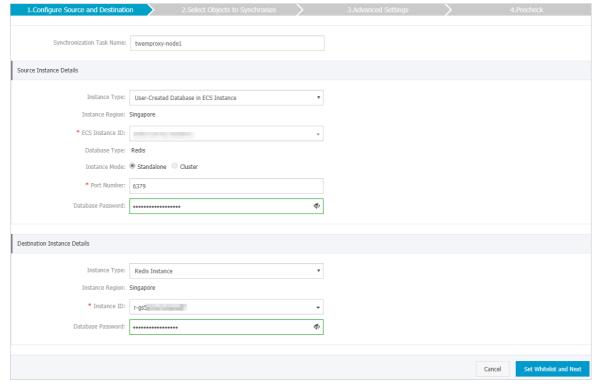
- If you run the EVAL or EVALSHA command to call Lua scripts, DTS cannot identify
 whether these Lua scripts are executed on the destination database. During
 incremental data synchronization, the destination database does not explicitly return
 the execution results of Lua scripts.
- When DTS calls the SYNC or PSYNC command to transfer data of the LIST type, DTS does not clear the existing data. In this case, the destination database may contain duplicate data records.

Procedure

- 1. Purchase a data synchronization instance. For more information, see Purchase procedure.
 - ? Note On the buy page, set Source Instance to Redis, Target Instance to Redis, and Synchronization Topology to One-Way Synchronization.
- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.



- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

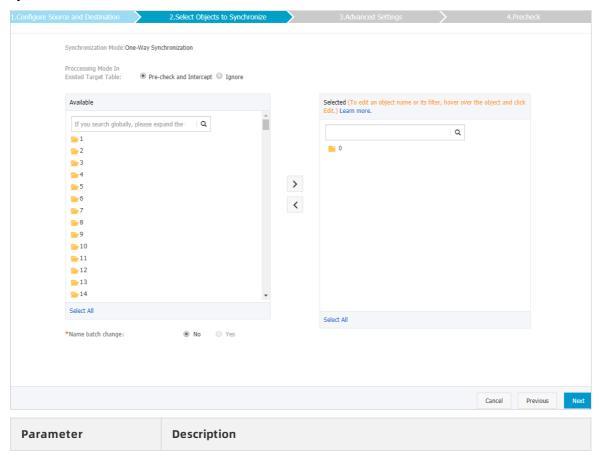


| Section | Parameter | Description |
|---------|-------------------------------|--|
| N/A | Synchronizati on Task Name | DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name. |
| | Instance Type | Select User-Created Database in ECS Instance. |
| | Instance Region | The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | | |

| Section | Parameter | Description |
|------------------------------------|----------------------|---|
| Source Instance Details | ECS Instance ID | Select the ID of the ECS instance where the master node of the Redis-Server resides. |
| | | Note DTS synchronizes each Redis-Server of the Twemproxy Redis cluster by using a data synchronization task till the whole cluster is synchronized. In this step, enter the ECS instance ID for the master node of Redis-Server 1. When you configure the data synchronization task for Redis-Server 2, enter the ECS instance ID for the master node of Redis-Server 2. You can configure data synchronization tasks for all Redis-Servers by following the procedure described in this topic. |
| | Database Type | The value of this parameter is set to Redis. |
| | | Select Standalone. |
| | Instance Mode | Note You must select Standalone for this parameter because data from a Twemproxy Redis cluster cannot be synchronized at a time. DTS synchronizes each Redis- Server of the cluster in a data synchronization task till all Redis-Servers are synchronized. |
| | Port Number | Enter the service port number of the master node in the Redis-Server. |
| | | Enter the database password for the master node. |
| | Database Password | Note This parameter is optional and can be left blank if no database password is set. |
| | Instance Type | Select Redis Instance. |
| | Instance Region | The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter. |
| | Instance ID | Select the ID of the destination ApsaraDB for Redis instance. |
| Destination Instance Details | | |

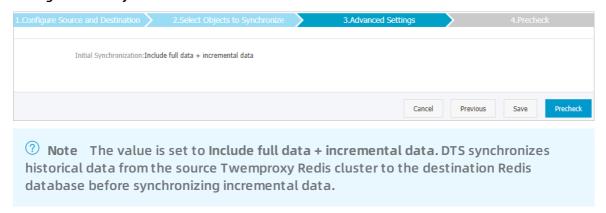
| Section | Parameter | Description |
|---------|----------------------|--|
| | Database Password | Enter the database password of the destination ApsaraDB for Redis instance. |
| | | ? Note The format of the database password is <user>:<password>. For example, if the username of the custom account is admin and the password is Rp829dlwa, the database password is admin:Rp829dlwa.</password></user> |

- 7. In the lower-right corner of the page, click Set Whitelist and Next.
 - Note The CIDR blocks of DTS servers are automatically added to the inbound rule of the ECS instance and the whitelist of the ApsaraDB for Redis instance. This ensures that DTS servers can connect to the source and destination instances.
- 8. Configure the processing mode in existing destination tables and the objects to be synchronized.



| Parameter | Description |
|--|--|
| Processing Mode In Existed Target Table | DTS synchronizes each Redis-Server of the Twemproxy Redis cluster in a data synchronization task till the whole cluster is synchronized. When you configure data synchronization for Redis-Server 1, if the ApsaraDB for Redis instance has no data, select Pre-check and Intercept. When you configure data synchronization for Redis-Server 2 to N, select Ignore. Otherwise, errors may occur during data synchronization. |
| | Note Pre-check and Intercept: checks whether the destination database is empty. If the destination database is empty, the precheck is passed. If the database is not empty, an error is returned during precheck and the data synchronization task cannot be started. Ignore: skips the precheck for empty destination databases and continues with data synchronization. If the keys in the destination database are the same as those in the source database during data synchronization, the keys in the source database overwrite those in the destination database. |
| Objects to be synchronized | Select databases from the Available section and click the icon to move the databases to the Selected section. You can select only databases as the objects to be synchronized. You cannot select keys as the objects to be synchronized. |

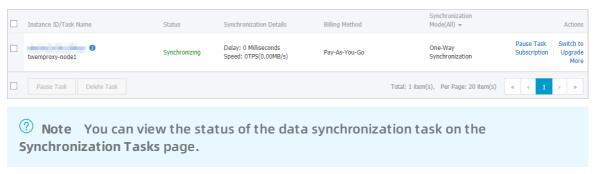
- 9. In the lower-right corner of the page, click Next.
- 10. Configure initial synchronization.



11. In the lower-right corner of the page, click Precheck.



- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.
- 12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.
- 13. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.



14. Create and configure a data synchronization task for every other Redis-Server by repeating steps 1 to 13.

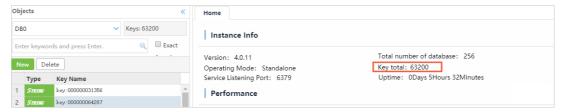
Result

In this topic, the Twemproxy Redis cluster consists of two Redis-Servers. You must create two data synchronization tasks. The following figure shows that the initial synchronization is complete for both tasks and both tasks are in the **Synchronizing** state.



In this topic, the database DBO is synchronized. You can use Data Management (DMS) to log on to the destination ApsaraDB for Redis instance and check the total number of keys in the ApsaraDB for Redis instance. The total number of keys is the same as that in the Twemproxy Redis cluster.

ApsaraDB for Redis instance



Source Twemproxy Redis cluster

```
root@ :~# redis-cli -p 6379 info|grep db0
db0:keys=29421,expires=0,avg_ttl=0
root@ :~# redis-cli -p 6389 info|grep db0
db0:keys=33779,expires=0,avg_ttl=0
root@iZbp1ib0ezn1xol5wbfsadZ:~#
```