Alibaba Cloud
Data Transmission Service

Data synchronization

Issue: 20200610

Legal disclaimer

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

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, 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 contact Alibaba Cloud directly if you discover any errors in this document.

Document conventions

Style	Description	Example
	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.
	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.
	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.
	A note indicates supplemental instructions, best practices, tips, and other content.	 **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings** > **Network** > **Set network type**.
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click **OK**.
Courier font	Courier font is used for commands.	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`

Style	Description	Example
{} or {a|b}	This format is used for a required value, where only one item can be selected.	switch {active|stand}

Contents

1 Overview of data synchronization scenarios

You can use DTS to synchronize data between homogeneous and heterogeneous data sources. DTS provides the following extract, transform, and load (ETL) capabilities: object name mapping for columns, tables, and databases, and data filtering. The data synchronization feature is applicable to scenarios such as active geo-redundancy, geo-disaster recovery, zone-disaster recovery, cross-border data synchronization, query load balancing, cloud BI systems, and real-time data warehousing. The following table lists the topics for configuring data synchronization in various scenarios.

Scenario	Topic
Synchronize data between MySQL databases	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 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 a MySQL database to a different type of database	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

Scenario	Topic
	Synchronize data from a user-created MySQL database hosted on ECS to Elasticsearch
	Synchronize data from an ApsaraDB RDS MySQL instance to a MaxCompute project
Synchronize data to or from a PolarDB cluster	Configure one-way data synchronization between Apsara PolarDB for MySQL clusters
	Synchronize data between PolarDB clusters compatible with Oracle
	Synchronize data from an Apsara PolarDB for MySQL cluster to an ApsaraDB RDS for MySQL instance
	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 Redis databases	Configure one-way data synchronization between ApsaraDB for Redis instances
	Synchronize data from a user-created Redis database hosted on ECS to an ApsaraDB for Redis instance
	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

2 Database types, initial synchronization types, and synchronization topologies

You can use DTS to synchronize data between two data sources in real time. This feature applies to multiple scenarios, such as active geo-redundancy, geo-disaster recovery, zone-disaster recovery, cross-border data synchronization, query load balancing, cloud BI systems, and real-time data warehousing. This topic describes the database types, initial synchronization types, and synchronization topologies that are supported by DTS.

For more information about data synchronization in various scenarios, see Overview of data synchronization scenarios.

Initial synchronization types

Initial synchronization type	Description
Initial schema synchronization	DTS synchronizes the schemas of the required objects from the source database to the destination database. Tables, views, triggers, and stored procedures can be synchronized. **Note:** Before you configure a data synchronization task, check whether initial schema synchronization is supported. If initial schema synchronization is not supported, you must create the destination database and tables based on the schemas of the required objects in the source database.
Initial full data synchronization	DTS synchronizes the historical data of the required objects from the source database to the destination database. The data is the basis for subsequent incremental synchronization. To simplify data synchronization, we recommend that you select both **initial schema synchronization** and **initial full data synchronization** when configuring a data synchronization task.

Synchronization topologies

For more information, see Synchronization topologies.

Database types, initial synchronization types, and synchronization topologies

A user-created source or destination database, such as a MySQL or Redis database, can be one of the following types:

- **User-Created Database in ECS Instance**
- **User-Created Database Connected Over Express Connect, VPN Gateway, or Smart Access Gateway**
- **Database without public IP:Port (Accessed through database gateway)**
- **Self built database accessed through Cloud Enterprise Network(CEN)**

Source database	Destination database	Initial synchronization type	Synchronization topology
• User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0 • ApsaraDB RDS for MySQL All versions	User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0	Initial schema synchronization Initial full data synchronization	One-way synchronization Two-way synchronization
	ApsaraDB RDS for MySQL All versions	Initial schema synchronization Initial full data synchronization	One-way synchronization Two-way synchronization

Source database	Destination database	Initial synchronization type	Synchronization topology
	Apsara PolarDB for MySQL All versions	Initial schema synchronization Initial full data synchronization	One-way synchronization
	Elasticsearch 5.5, 6.3, and 6.7	Initial schema synchronization Initial full data synchronization	One-way synchronization
	MaxCompute All versions	Initial schema synchronization Initial full data synchronization	One-way synchronization
	AnalyticDB for MySQL 2.0 and 3.0	Initial schema synchronization Initial full data synchronization	One-way synchronization

Source database	Destination database	Initial synchronization type	Synchronization topology
	AnalyticDB for PostgreSQL (Previous name: HybridDB for PostgreSQL) 4.3 and 6.0	Initial schema synchronization Initial full data synchronization	One-way synchronization
Apsara PolarDB for MySQL All versions	Apsara PolarDB for MySQL All versions	Initial schema synchronization Initial full data synchronization	One-way synchronization
	ApsaraDB RDS for MySQL All versions	Initial schema synchronization Initial full data synchronization	One-way synchronization
	User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0	Initial schema synchronization Initial full data synchronization	One-way synchronization

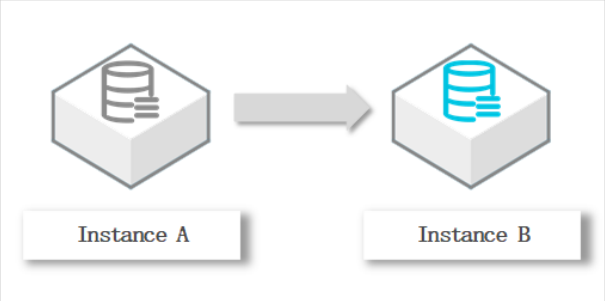
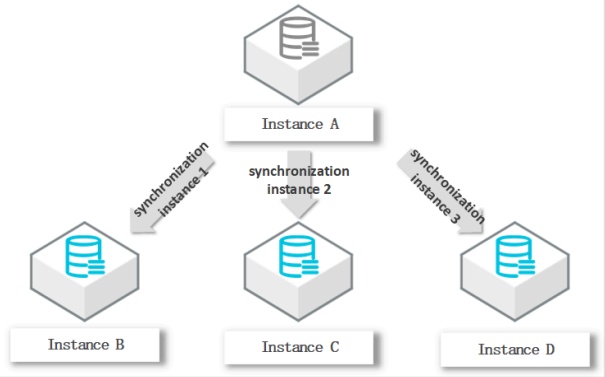
Source database	Destination database	Initial synchronization type	Synchronization topology
	AnalyticDB for MySQL 2.0 and 3.0	Initial schema synchronization Initial full data synchronization	One-way synchronization
• User-created Redis database (single-host and cluster architecture) 2.8, 3.0, 3.2, 4.0, and 5.0 • ApsaraDB for Redis instance (single-host and cluster architecture) Community versions 4.0 and 5.0	• User-created Redis database (single-host and cluster architecture) 2.8, 3.0, 3.2, 4.0, and 5.0 • ApsaraDB for Redis instance (single-host and cluster architecture) Community versions 4.0 and 5.0	Initial full data synchronization **Note:** Redis is a NoSQL database that does not require initial schema synchronization.	One-way synchronization

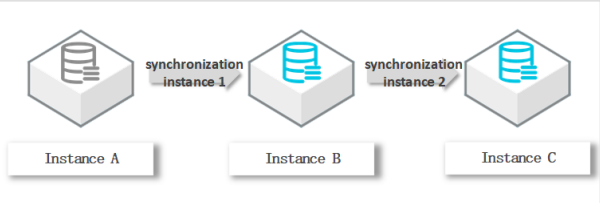
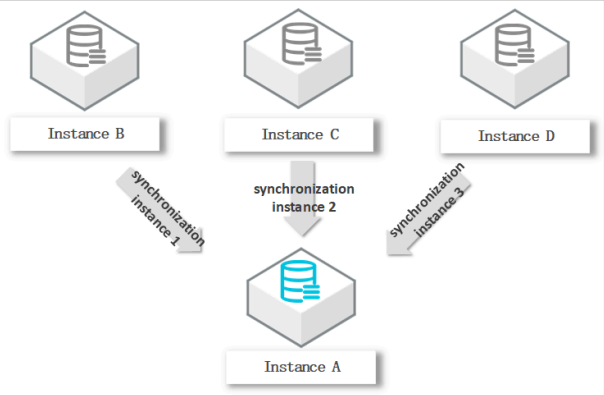
3 Synchronization topologies

The data synchronization feature supports multiple types of synchronization topologies. You can plan your synchronization instances based on your business requirements.

One-way synchronization

To ensure data consistency for one-way synchronization, we recommend that you perform only read operations on the objects in the destination instance. Do not modify the objects.

Topology type	Topology	Description
One-way one-to-one synchronization		None
One-way one-to-many synchronization		You must purchase multiple synchronization instances to implement one-way one-to-many synchronization. For example, if you want to synchronize data from Instance A to Instance B, C, and D, you must purchase three synchronization instances.

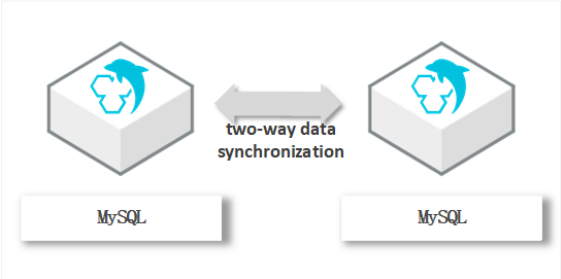
Topology type	Topology	Description
One-way cascade synchronization		You must purchase multiple synchronization instances to implement one-way cascade synchronization. For example, if you want to synchronize data from Instance A to Instance B and then from Instance B to Instance C, you must purchase two synchronization instances.
One-way many-to-one synchronization		You must purchase multiple synchronization instances to implement one-way many-to-one synchronization. For example, if you want to synchronize data from Instance B, C, and D to Instance A, you must purchase three synchronization instances. **Note:** To ensure data consistency, you must select different objects for these synchronization instances.

Two-way synchronization

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

Note:

- For more information about how to configure two-way synchronization, 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 #unique_32.

Topology type	Topology	Description
Two-way one-to-one synchronization		To ensure data consistency, make sure that records with the same primary key, business primary key, or unique key are updated only on one of the instances.

References

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

4 Data type mappings for initial schema synchronization

The data types of different databases do not have one-to-one correspondence. During initial schema synchronization, DTS maps the data types of the source database to those of the destination database. The following tables list the data type mappings for you to view and evaluate the impact of data synchronization on your business.

Synchronize data from MySQL/ApsaraDB RDS for MySQL/PolarDB for MySQL to AnalyticDB for MySQL

MySQL data type	AnalyticDB for MySQL data type
bigint unsigned	decimal(20,0)
binary	VARBINARY
bit	VARCHAR
blob	VARBINARY
char	varchar
date	DATE
datetime	DATETIME
decimal	decimal
double	double
enum	VARCHAR
FLOAT	FLOAT
geometry	VARBINARY
geometrycollection	VARBINARY
int unsigned	BIGINT
integer	int
json	JSON
linestring	VARBINARY
LONGBLOB	VARBINARY
LONGTEXT	varchar

MySQL data type	AnalyticDB for MySQL data type
MEDIUMBLOB	VARBINARY
mediumint	int
mediumint unsigned	INT
MEDIUMTEXT	varchar
multilinestring	VARBINARY
multipoint	VARBINARY
multipolygon	VARBINARY
numberic	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

Synchronize data from MySQL/ApsaraDB RDS for MySQL to Elasticsearch

MySQL data type	Elasticsearch data type	Description
BIGINT	LONG	None
BINARY	BINARY	None
BIT	LONG	If the data is only one byte long, we recommend that you use the BOOLEAN type in Elasticsearch.

MySQL data type	Elasticsearch data type	Description
BLOB	BINARY	None
BOOL or BOOLEAN	BOOLEAN	None
CHAR	TEXT	None
DATE	DATE	The DATE format is yyyy-MM-dd, in UTC. For more information, see Date format mappings.
DATETIME	DATE	The DATE format is yyyy-MM-dd'T'HH:mm:ss, in UTC. If DATE is accurate to microseconds, its format is 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 to ensure data consistency.
DOUBLE	DOUBLE	None
FLOAT	FLOAT	None
GEOMETRY	GEO_SHAPE	None
GEOMETRYCOLLECTION	GEO_SHAPE	None
INT	INTEGER	If the data type of the source database is UNSIGNED INT, we recommend that you use the LONG type in Elasticsearch.
JSON	OBJECT	If the data is only one byte long, we recommend that you use the BOOLEAN type in Elasticsearch.
LINESTRING	GEO_SHAPE	None
LONGTEXT	TEXT	None
MEDIUMINT	INTEGER	None
MEDIUMTEXT	TEXT	None
MULTILINESTRING	GEO_SHAPE	None
MULTIPOINT	GEO_SHAPE	If the data is only one byte long, we recommend that you use the BOOLEAN type in Elasticsearch.
MULTIPOLYGON	GEO_SHAPE	None

MySQL data type	Elasticsearch data type	Description
POINT	GEO_POINT	None
POLYGON	GEO_SHAPE	None
REAL	DOUBLE	None
SMALLINT	SHORT	If the data type of the source database is UNSIGNED SMALLINT, we recommend that you use the INTEGER type in Elasticsearch.
TIME	DATE	The DATE format is HH:mm:ss, in UTC. If DATE is accurate to microseconds, its format is 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. If DATE is accurate to microseconds, its format is yyyy-MM-dd'T'HH:mm:ss.S. For more information, see Date format mappings.
TINYINT	SHORT	If the data type of the source database is UNSIGNED TINYINT, we recommend that you use the INTEGER type in Elasticsearch.
TINYTEXT	TEXT	None
VARCHAR	TEXT	None
YEAR	DATE	The DATE format is yyyy, in UTC. For more information, see Date format mappings.

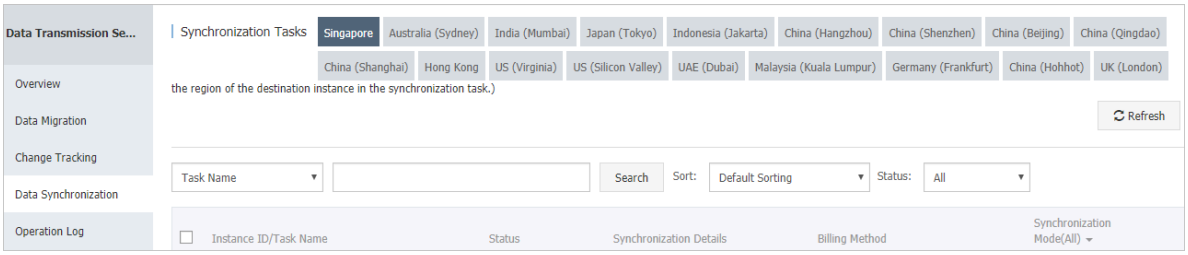
5 Synchronization task management

5.1 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.

Note:

You can view the statistics during the last 15 days.

Table 5-1: Performance metrics

Performance metric	Description
Bandwidth	The bandwidth between the data writing module and the data pulling module. Unit: MB/s.
Synchronization speed (TPS)	The number of transactions that DTS synchronizes to the destination database per second.
Synchronization 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.

5.2 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.
- When you modify a data synchronization task, the source and destination databases must be running. The source and destination databases cannot be in the process of upgrade, configuration change, network switchover, or cross-zone migration. This

ensures that DTS can read database and table information from the source database and connect to the source and destination databases.

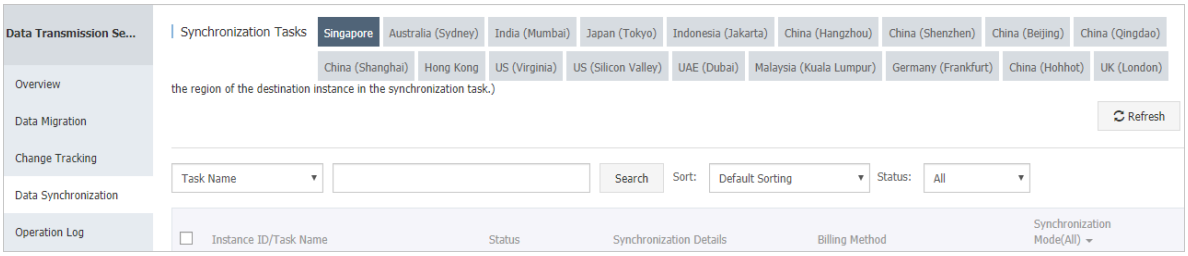
Notes

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

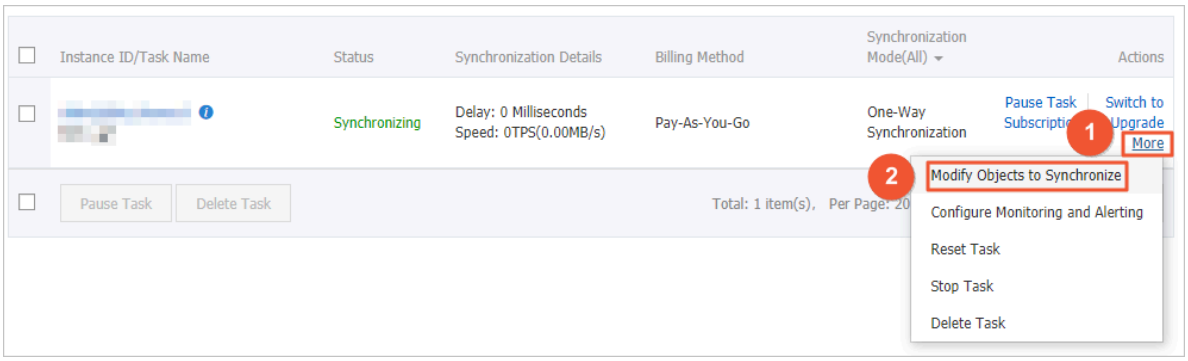
- If initial synchronization is not configured for the data synchronization task, DTS performs data synchronization for the new object after incremental data is generated on the source instance.
- If initial synchronization is configured for the data synchronization task, DTS performs initial schema synchronization and initial full data synchronization before synchronizing incremental data.

Procedure

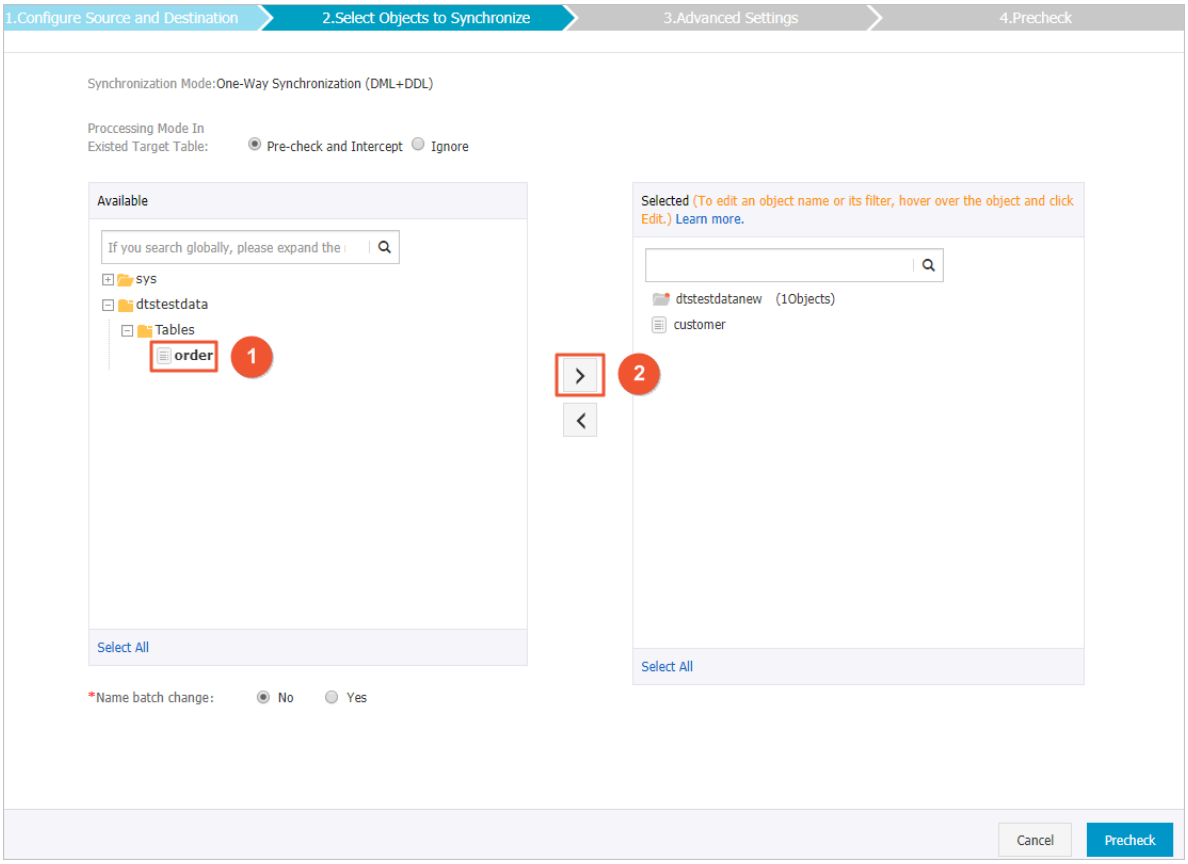
1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Synchronization**.
3. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



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 and Start**.

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 the precheck, click the icon next to each failed item to view details. Fix the issues as instructed and run the precheck again.

7. Start the data synchronization task.

5.3 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

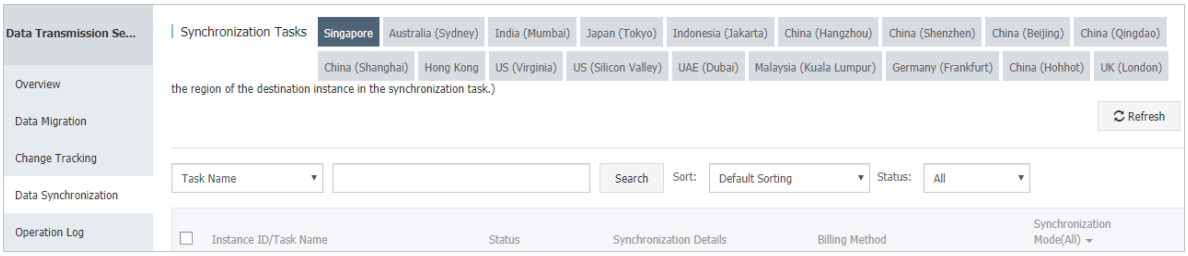
- The data synchronization task is in the **Synchronizing**, **Paused**, or **Synchronization Failed** state.
- When you modify a data synchronization task, the source and destination databases must be running. The source and destination databases cannot be in the process of upgrade, configuration change, network switchover, or cross-zone migration. This ensures that DTS can read database and table information from the source database and connect to the source and destination databases.

Notes

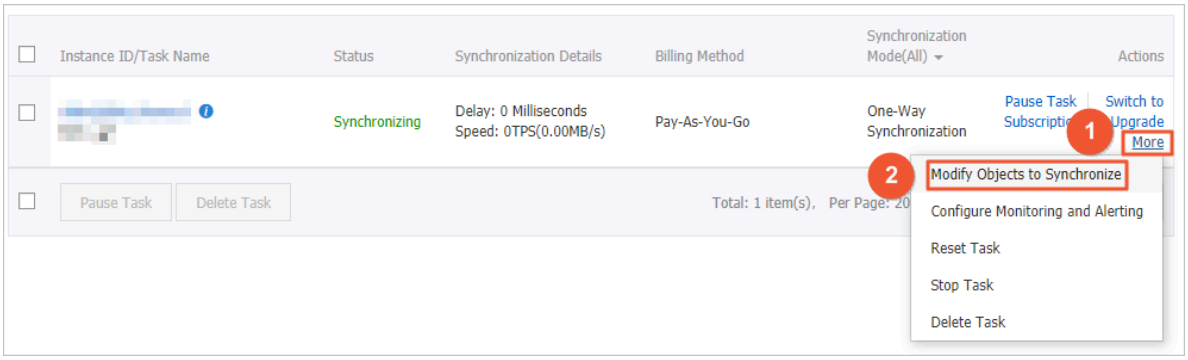
After an object is removed from the data synchronization task, the incremental data of the object is no longer synchronized to the destination database.

Procedure

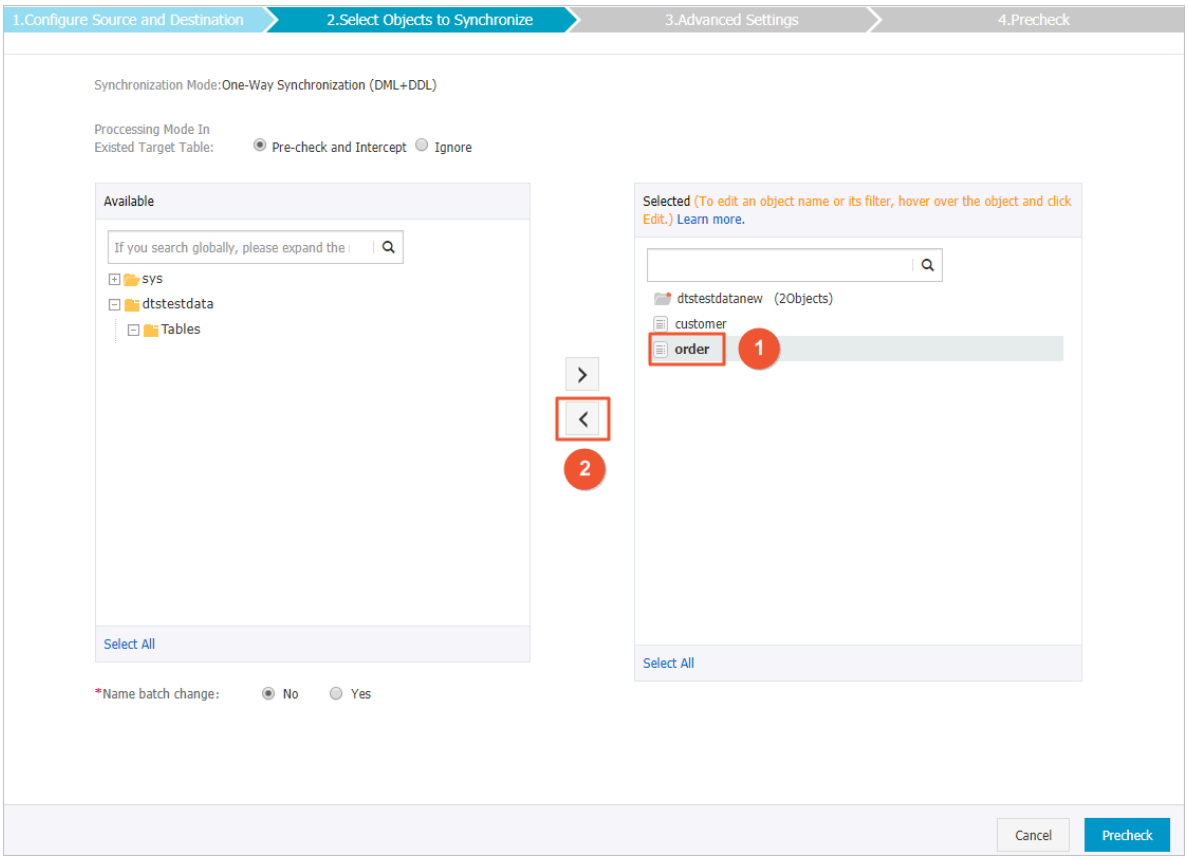
1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Synchronization**.
3. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



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 icon to move the object to the **Available** section.



6. Click **Precheck and Start**.

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 the precheck, click the icon next to each failed item to view details. Fix the issues as instructed and run the precheck again.

7. Start the data synchronization task.

5.4 Specify the name of an object in the destination instance

After an object, such as a database or table, is synchronized from the source instance to the destination instance, the name of the object remains unchanged. You can use the object name mapping feature provided by DTS to specify a different name for the object in the destination instance.

Notes

You can perform this operation only when a data synchronization task is configured and the current process is **Select Objects to Synchronize**.

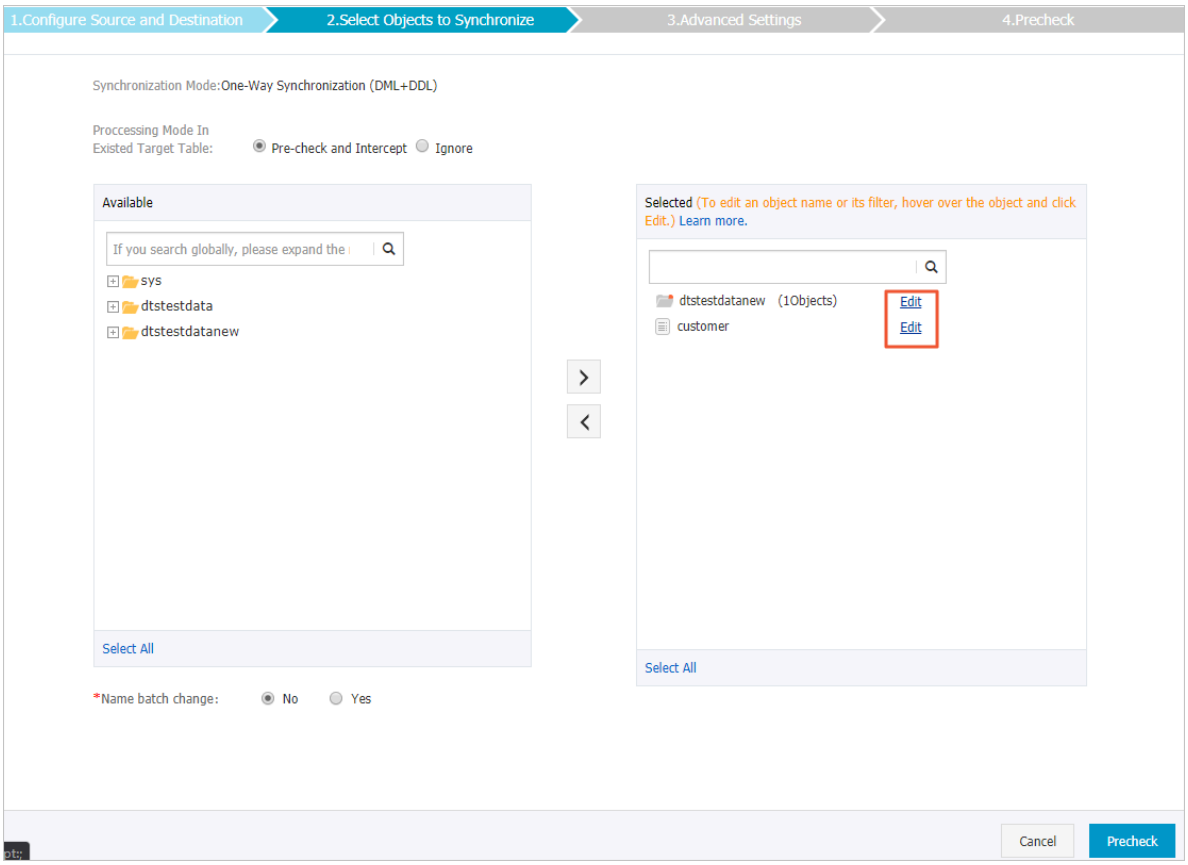
Note:
Do not perform this operation after the data synchronization task is started. Otherwise, the synchronization may fail.

Procedure

1. On the **Select Objects to Synchronize** page, move the required objects to the **Selected** section, move the pointer over a database or table, and then click **Edit**.

   Note:

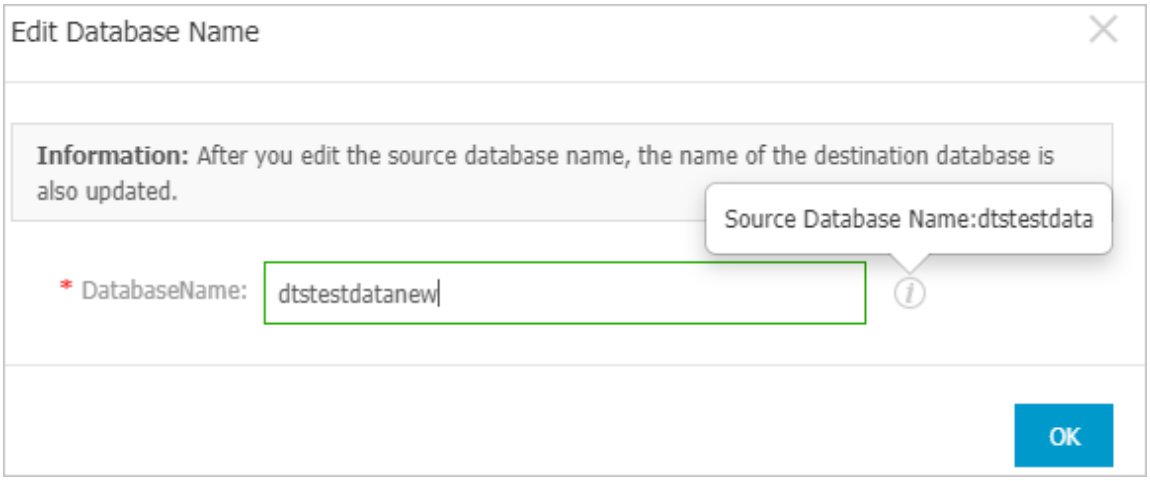
Different database types support different objects. If **Edit** appears when you move the pointer over the target object, the operation is supported.



2. In the dialog box that appears, specify a name for the object in the destination instance.

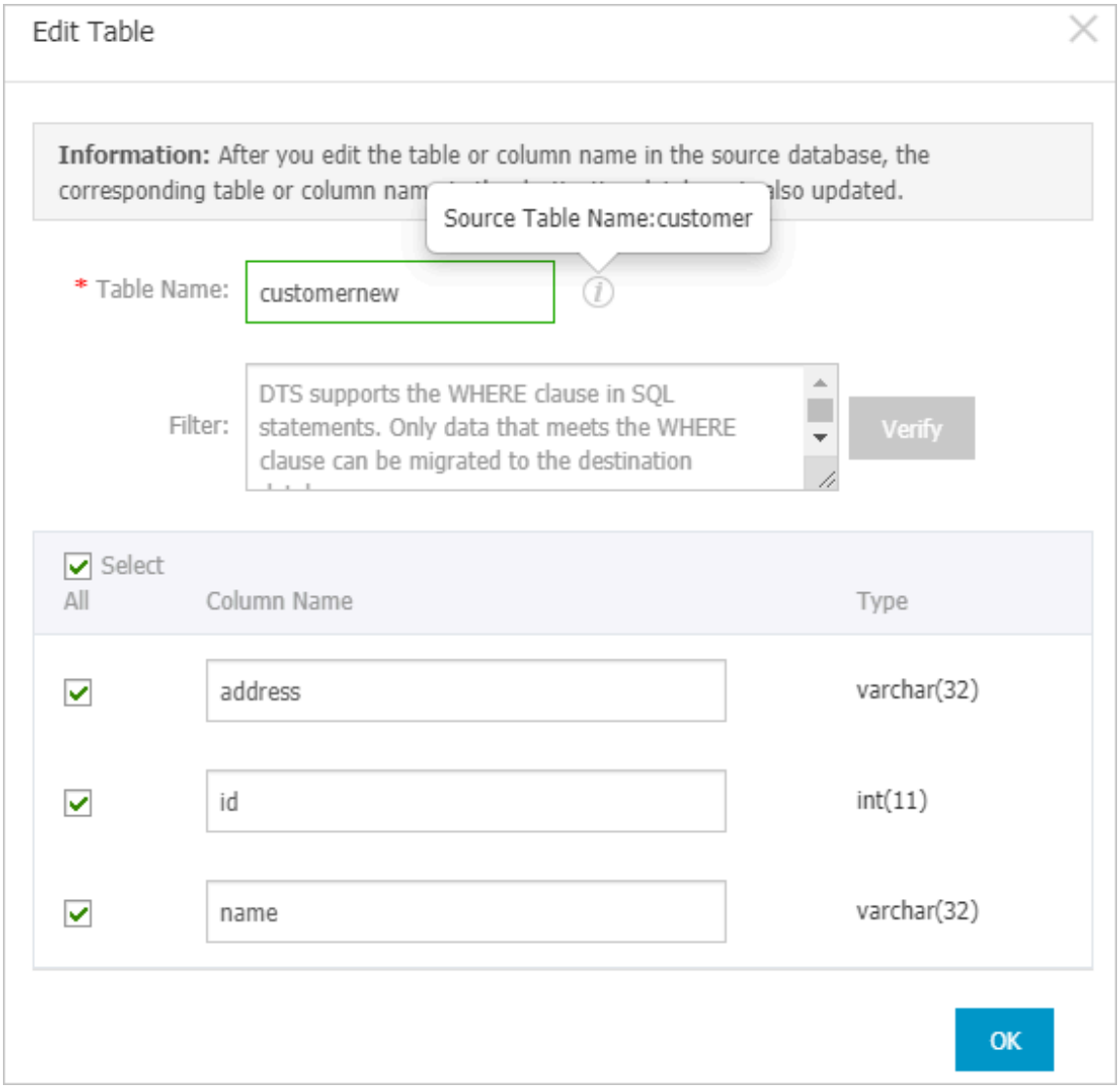
   - Database name mapping

     In the **Edit Database Name** dialog box that appears, enter the database name that you want to use in the destination instance.

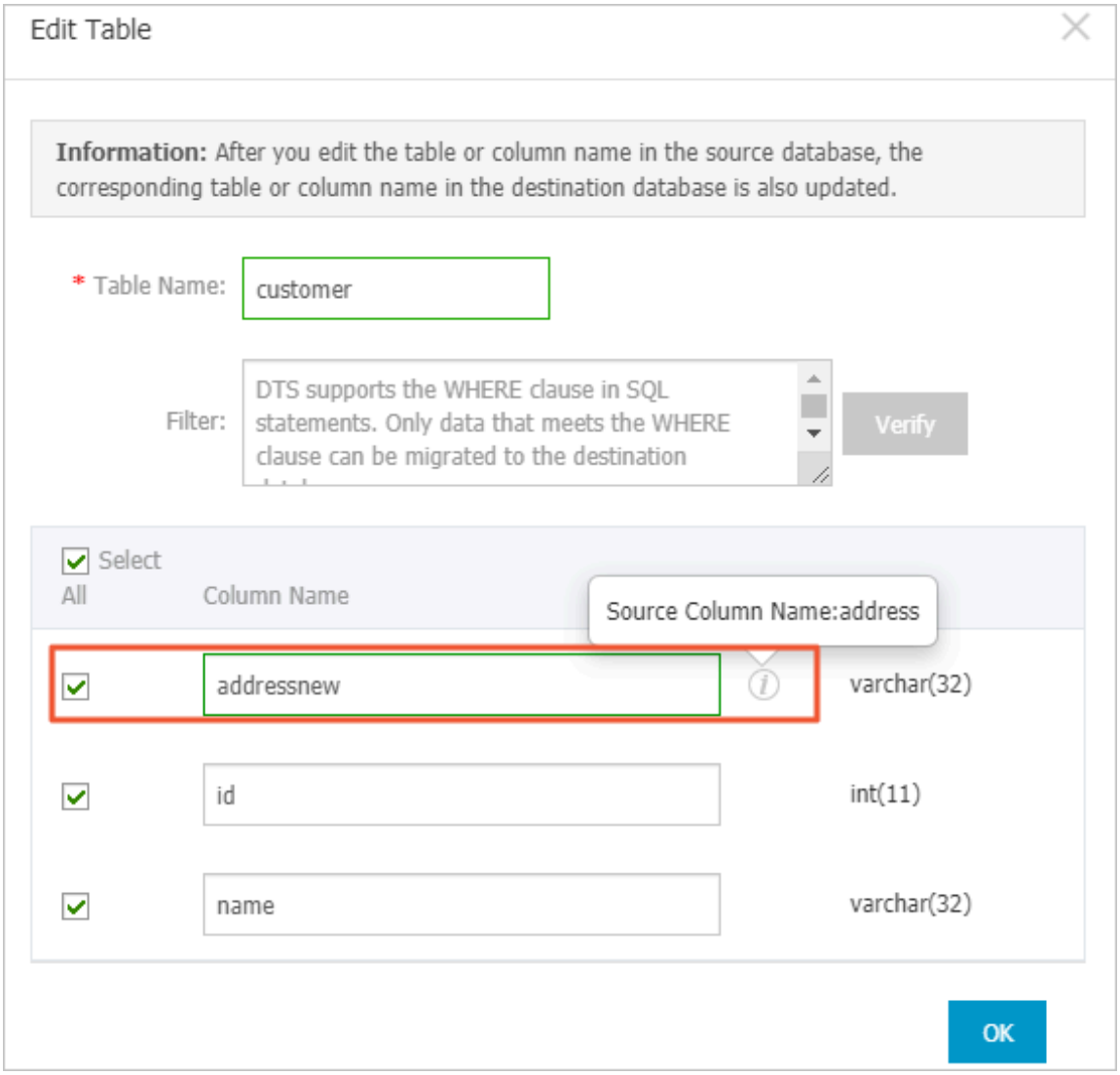
   - Table name mapping

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



- Column name mapping

  In the **Edit Table** dialog box that appears, enter a new name for each column.



Note:
In this step, you can deselect columns that do not need to be synchronized.

3. Click **OK**.
4. Configure other parameters that are required for the data synchronization task.

5.5 Use SQL conditions to filter data

When configuring the objects to be synchronized in a data synchronization task, you can specify SQL conditions to filter data. Only data that meets the specified conditions is

synchronized to the destination database. This feature is applicable to scenarios such as regular data synchronization and table partitioning.

Prerequisites

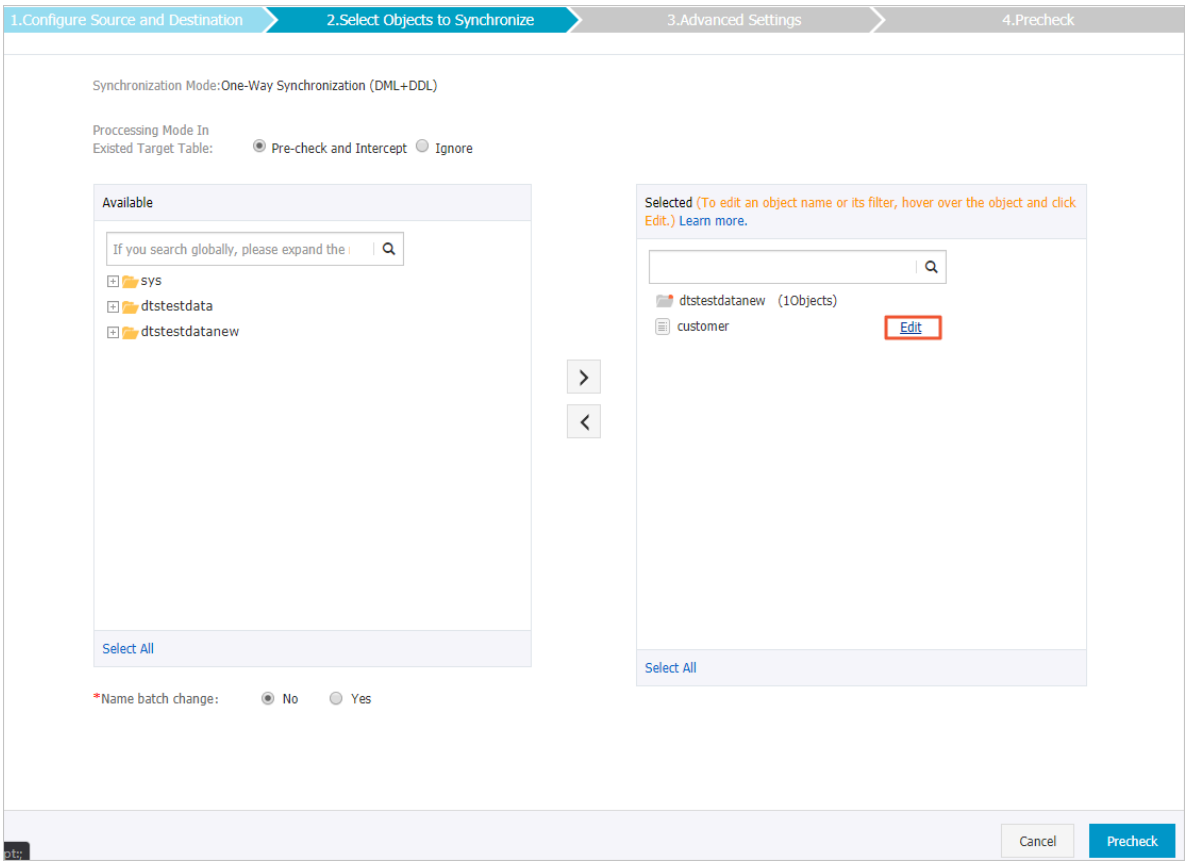
A data synchronization task is configured. The current step is **Select Objects to Synchronize**. For more information about how to create and configure a data synchronization task, see #unique_40.

Limits

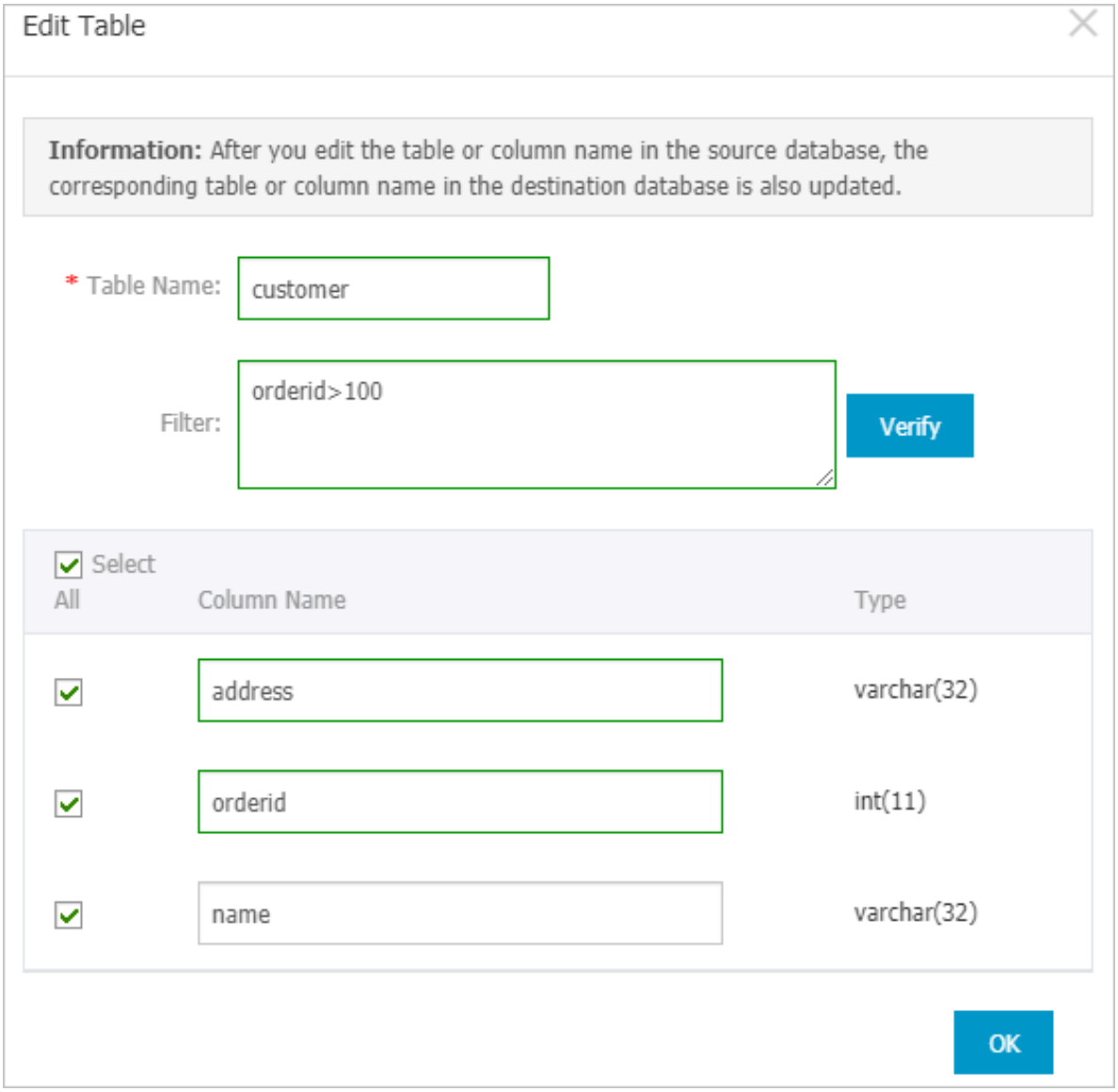
You can only filter the fields in the current table. Cross-table filtering is not supported.

Procedure

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



2. In the **Edit Table** dialog box that appears, enter an SQL condition in the **Filter** field.



Note:

- An SQL condition is a standard SQL WHERE statement. The following operators are supported: =, ! =, <, and >. Only data that meets the WHERE condition is migrated to the destination database. In this example, enter orderid>100.
- You can use apostrophes (') in an SQL condition if necessary. For example, you can enter address in('hangzhou','shanghai').

3. Click **Verify** to check whether the syntax is valid.

Note:

- If the syntax is valid, the **Information** message prompts that **the validation is passed**.

- If the syntax is invalid, the **Error** message appears and you must modify the SQL WHERE statement based on the instructions.

4. Click **OK**.
5. Configure other parameters that are required for the data synchronization task.

5.6 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

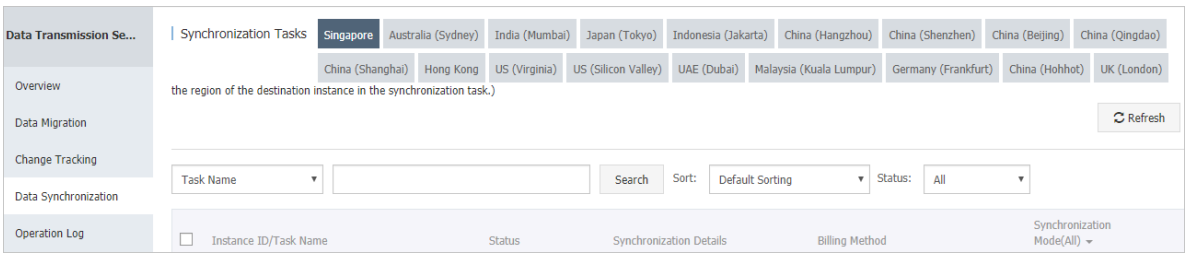
The data synchronization task is in the **Synchronizing**, **Paused**, **Synchronization Failed**, **Performing Initial Synchronization**, or **Initial Synchronization Failed** state.

Precautions

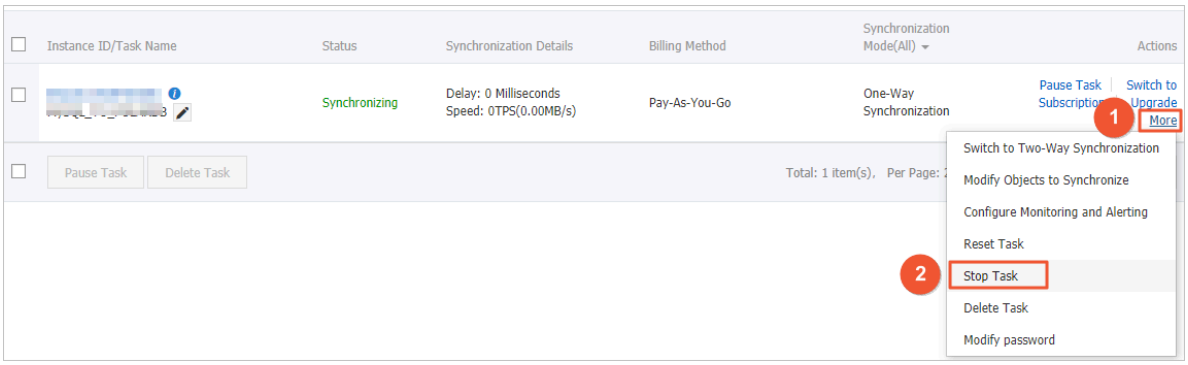
If a data synchronization task is stopped, the task enters the **Completed** state and no longer synchronizes incremental data from the source database to the destination database.

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

#unique_42

5.7 Reset a data synchronization task

This topic describes how to reset data synchronization task. You can reset a data synchronization task to stop data synchronization or reconfigure data synchronization.

Prerequisites

The data synchronization task is not in the **Not Configured** status.

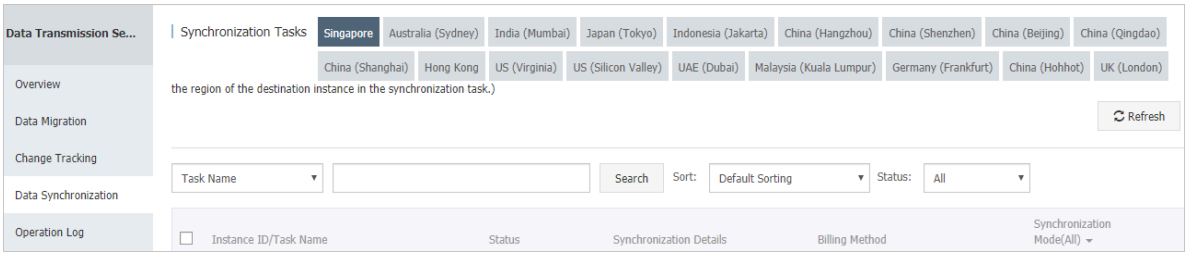
Impacts on billing

- Subscription: no impact.
- Pay-as-you-go: The data synchronization task will enter the **Not Configured** state. 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 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 Data Synchronization. Incremental data changes and schema changes in the source databases will not be synchronized to the destination databases.
> - The configuration of data synchronization task will be cleared and changed to **Not Configured** status.

5. In the dialog box that appears, click **OK**.

References

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

5.8 Solutions for the synchronization task pre-check failure

Before performing data synchronization task, DTS performs a precheck on the source and destination databases. This topic describes the possible causes for the failed precheck of various items and solutions to them.

Precheck items	Possible causes of failure	Solution
Source Database Connectivity Destination Database Connectivity	The database account or password is incorrect.	Obtain the correct password and modify the database password of the source or destination database configured in the synchronization task. For more information, see #unique_45.

Precheck items	Possible causes of failure	Solution
	Source or destination database has access restrictions on source IP addresses.	• If the source or destination database is MySQL, re-authorize the database account. The authorized IP address list must include the IP addresses of DTS servers. For more information about how to select the IP address of the DTS server, see #unique_46. • If the source or destination database is SQL Server database, disable the firewall or disable triggers. • If the source or destination database is Oracle, Modify `TCP.VALIDNODE_CHECKING` the value to `no` and restart the process.
	A firewall is configured for the server to which the source or destination database belongs.	Turn off firewall.
	Network interworking problem between DTS server and source or target database.	You can submit a ticket to contact Alibaba Cloud engineers for assistance.
Source Database Version	DTS does not support databases of the specified version. For more information about the database versions supported by DTS, see Database types, initial synchronization types, and synchronization topologies.	Upgrades or downgrades the database version.

Precheck items	Possible causes of failure	Solution
Destination Database Availability	When the destination databases does not exist in the destination instance, DTS would automatically creates the databases, the creation may fail due to the following reasons: • The database name contains special characters other than lowercase letters, digits, underscores (_), or hyphens (-). • The character set of the database is other than utf8, GBK, latin1, or utf8mb4. • The destination database account configured in the data synchronization task does not have correspond ing permissions.	Manually create a database that meets the requirements in the target instance or adjust the permissions of the database account.
Source Database Permissions Destination Database Permissions	The database account configured in data synchronization task does not have sufficient permissions. **Note:** The permissions required by each type of database are different. For more information, see the following cases: Overview of data synchronization scenarios.	Adjust the permissions of a database account.

Precheck items	Possible causes of failure	Solution
Schema Name Conflict	The object, such as database and table, has the same name as that in the target database.	• Log on to the target database and modify the database or table names. • Map the objects to be synchronized to objects with different names in the destination database by using the object name mapping feature provided by DTS. For more information, see Specify the name of an object in the destination instance. • Remove the object that has the same name as the object to be synchronized. For more information, see Remove an object from a data synchronization task.
Value of server_id in Source Database	The value of server-id is not a integer greater than or equal to 2.	Log on to the source database and modify the value of server-id . For more information, see #unique_47.
Source Database Binary Logging	The binlogging feature is not enabled for the source database.	Log on to the source database and enable binlogging. For more information, see #unique_48/unique_48_Connect_42_section_bls_hx5_thb.
Binlog Format of Source Database	The Binlog format is not set to ROW for the source database.	Log on to the source database and run set global binlog_format ='ROW ';, and then restart the MySQL process.

Precheck items	Possible causes of failure	Solution
Integrity constraint	The Parent Tables of the Child Tables are not selected as the objects to be synchronized. This impairs the integrity of foreign key constraints.	• Delete foreign key dependencies from the tables that fail the PreCheck. • Add the Parent Table to the object to be synchronized. For more information, see Add an object to a data synchronization task. • Removes the corresponding Child Tables from the objects to be synchronized. For more information, see Remove an object from a data synchronization task.
Storage Engine	The storage engine of the table to be synchronized is FEDERATED, MRG_MyISAM, or TokuDB.	Log on to the source database, and set the storage engine of the table to InnoDB.
Character Set	DTS does not support the character set of the objects to be synchronized (such as UCS2).	Log on to the source database , and set the character set of tables to utf8, GBK, latin1, or utf8mb4.
Complex topologies	DTS does not support the topology consisting of source and destination instances. For more information about the topologies supported by DTS and relevant precautions, see Synchronization topologies.	Stop the task or wait for the conflict to complete before performing the data synchronization.
Password Format of MySQL Database	Specify whether you set up the password with an old format for the source database.	Change the format of the database password. For more information, see 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

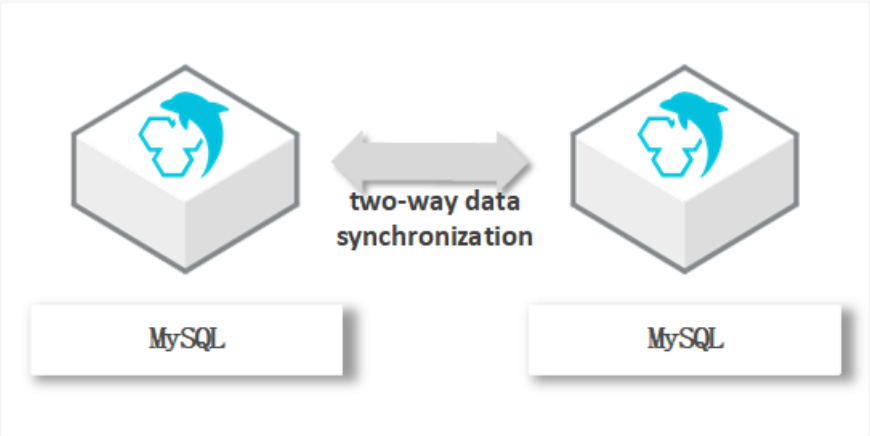
The source and destination ApsaraDB RDS for MySQL instances are created. For more information, see Create an RDS for MySQL instance.

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 • User-created database connected over a database gateway • Self built database accessed through Cloud Enterprise Network(CEN)	• 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 • Self built database accessed through Cloud Enterprise Network(CEN)

Operations that can be synchronized

Operation 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 gh-ost 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

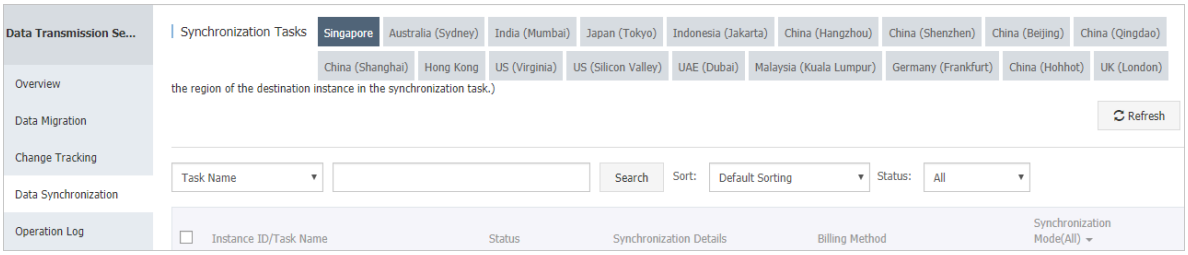
  To ensure the stability of a two-way synchronization channel, you can synchronize DDL updates of a single table only in one direction. If DDL synchronization in a direction is configured, DDL synchronization in the opposite direction is not supported. Only DML operations can be synchronized in the opposite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

Procedure

1. Purchase an instance for two-way data synchronization. For more information, see #unique_51/unique_51_Connect_42_section_39h_fto_gdl.

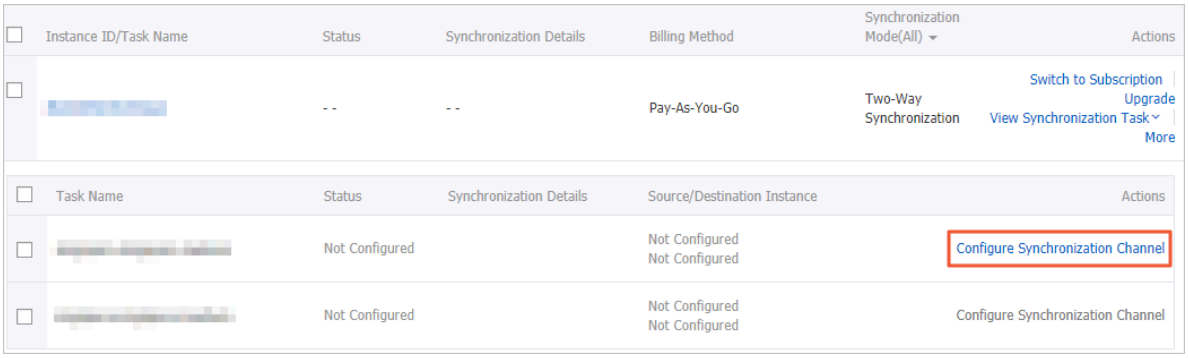
   Notice:
   On the buy page, set both Source Instance and Target Instance to **MySQL** and set Synchronization Topology to **Two-Way Synchronization**.

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.

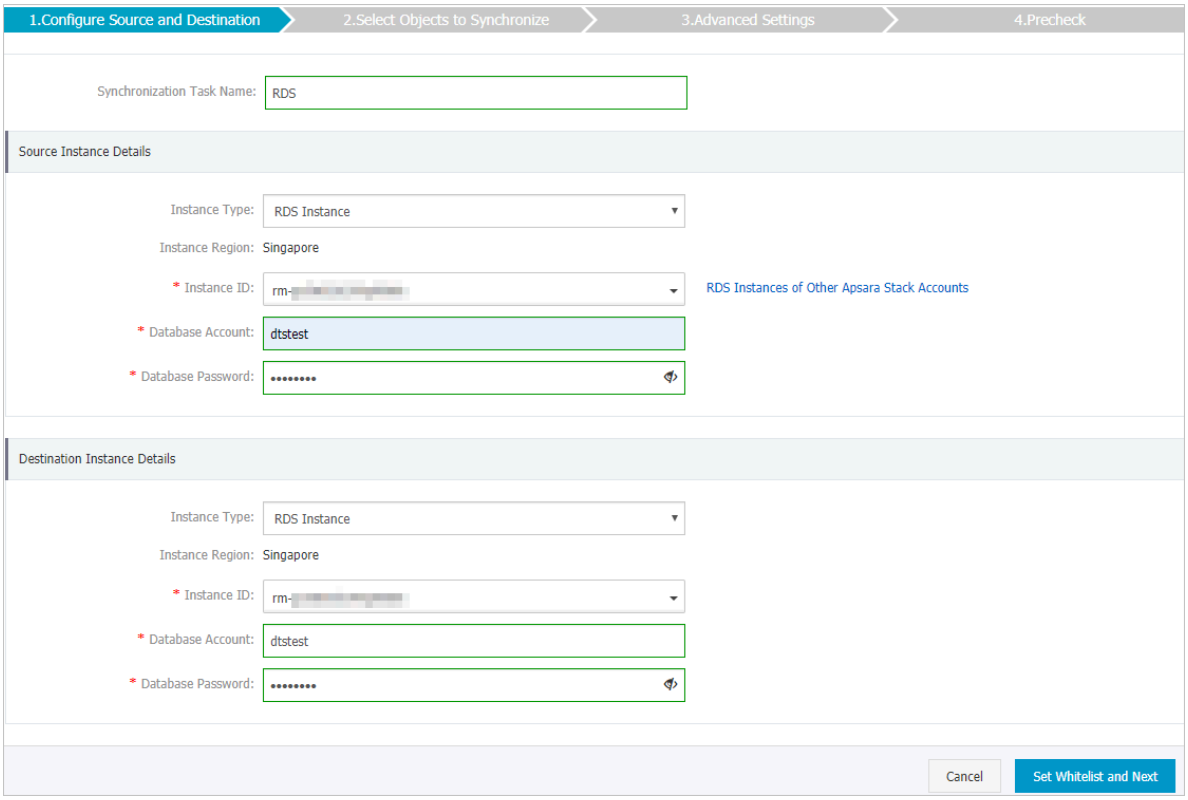


5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column of the first data synchronization task.

   Notice:
   A two-way data synchronization instance contains two data synchronization tasks. You must configure a channel for each task.



6. Configure the source and destination instances.



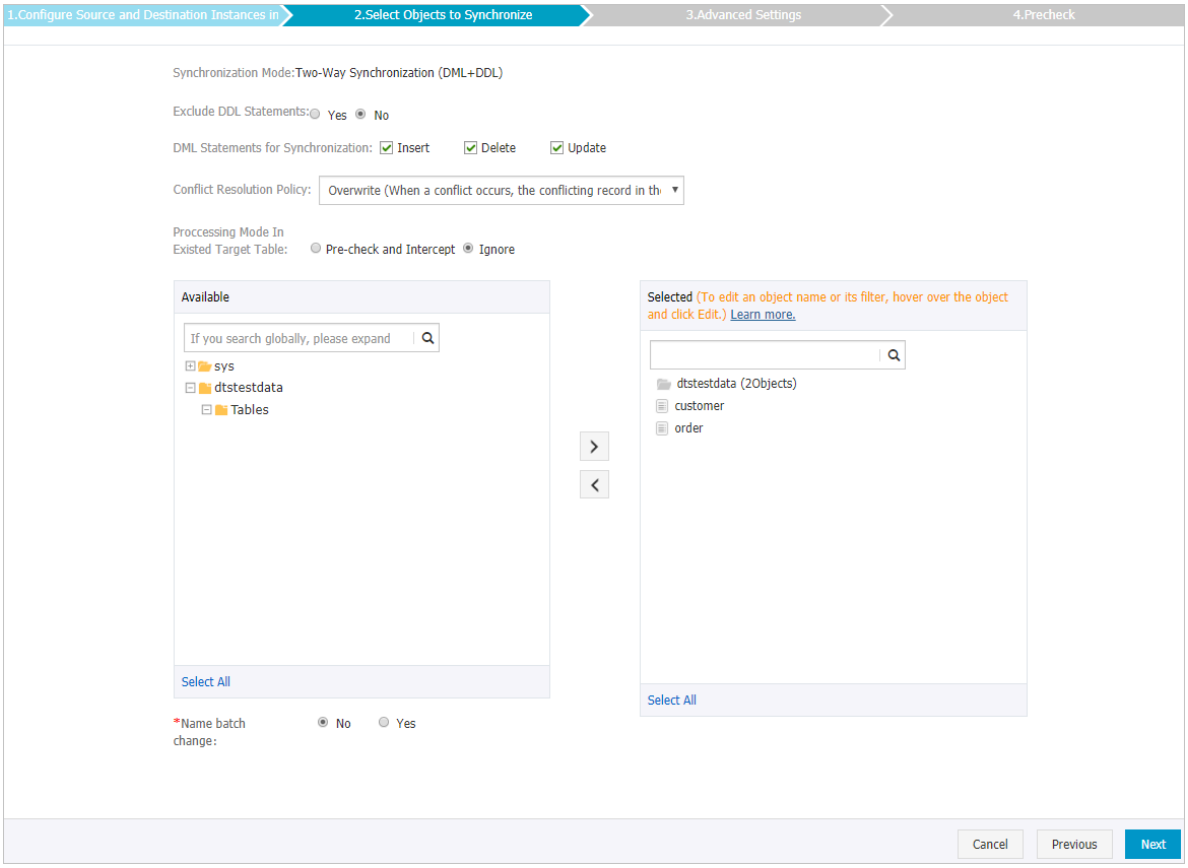
Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **RDS Instance**.
	Instance Region	The 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.
	Database Account	Enter the database account of the source RDS instance. **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**.

Section	Parameter	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.
Destination Instance Details	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.
	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.

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 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.



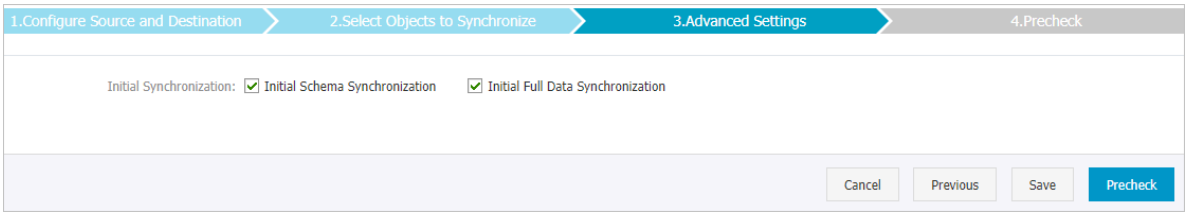
Section	Parameter	Description
Synchronization policy	Exclude DDL Statements	• To exclude DDL operations, select **Yes**. • To include DDL operations, select **No**. **Notice:** After you select No, the table does not support synchronizing DDL operations in the opposite direction.
	DML Statements for Synchronization	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.

Section	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. **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.

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.

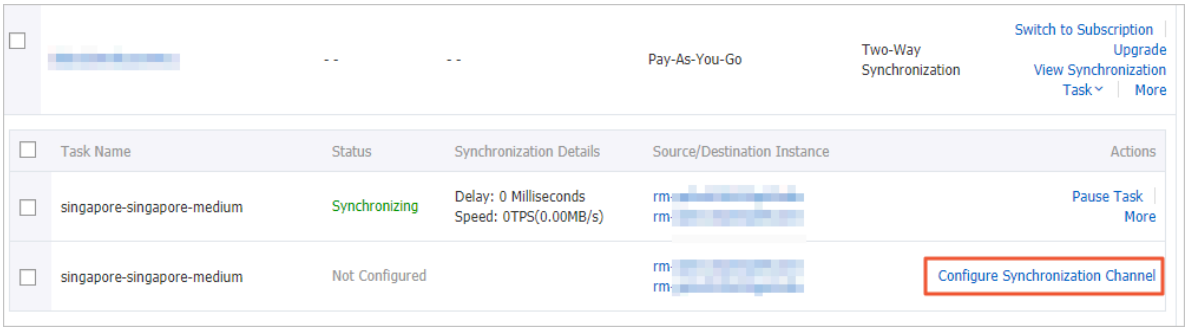
11. In the lower-right corner of the page, click **Precheck**.

Notice:

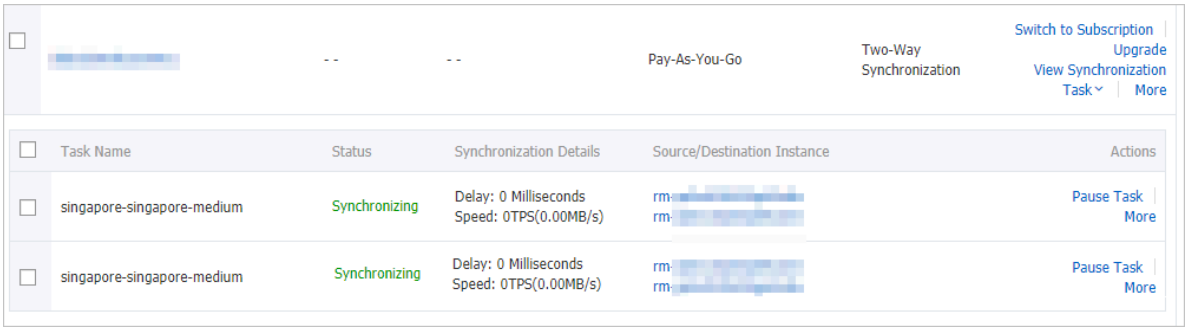
- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes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

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE

Operation 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

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.

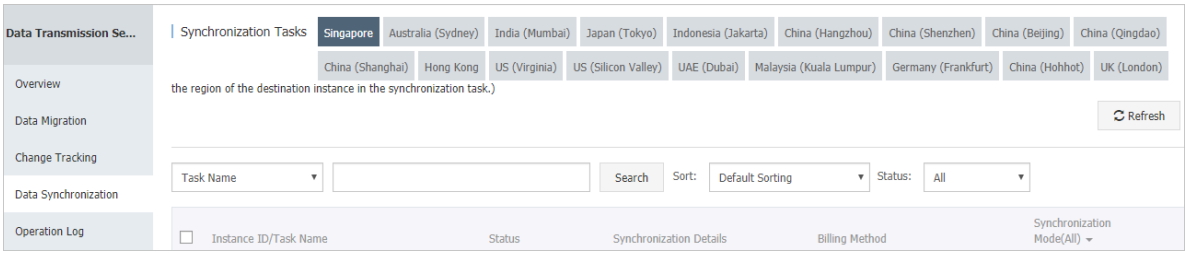
Procedure

1. Purchase an instance for one-way data synchronization. For more information, see #unique_51/unique_51_Connect_42_section_39h_fto_gdl.

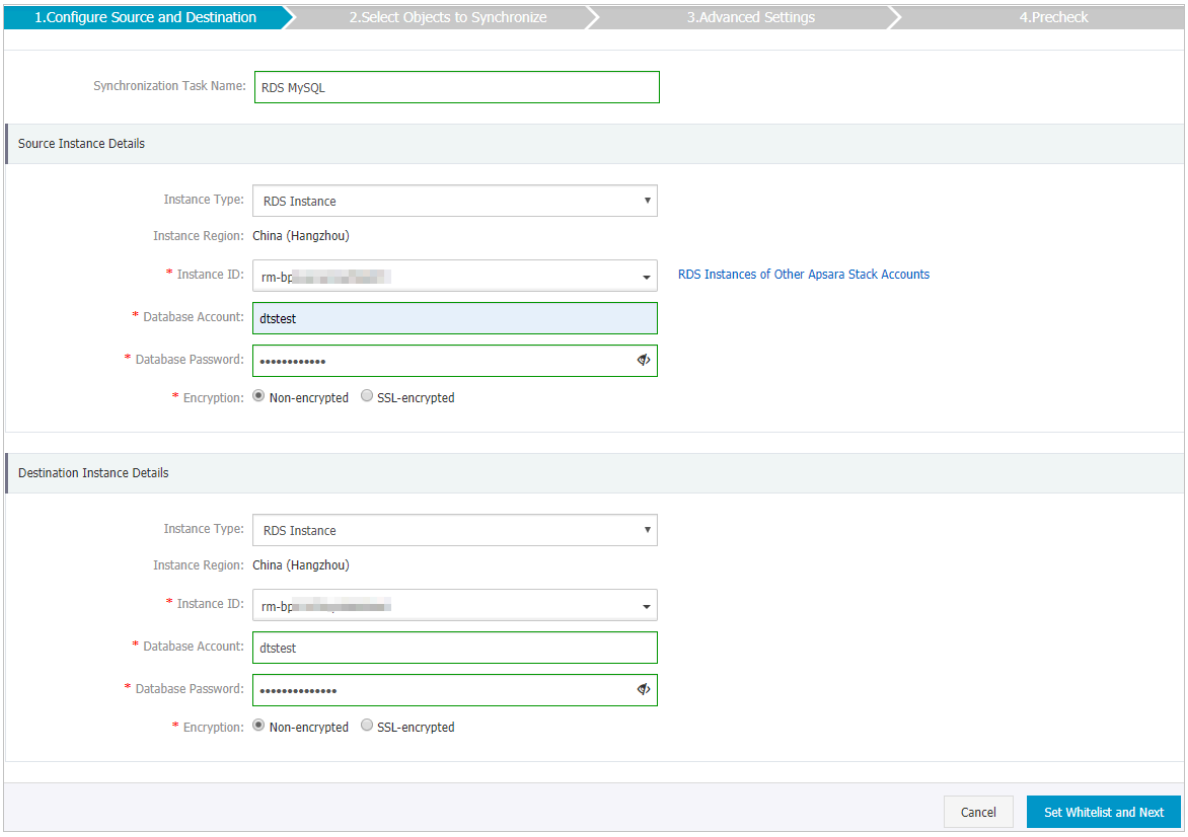
   Note:
   Select **MySQL** for both the source instance and the destination instance and select **One-Way Synchronization** as the synchronization topology.

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.



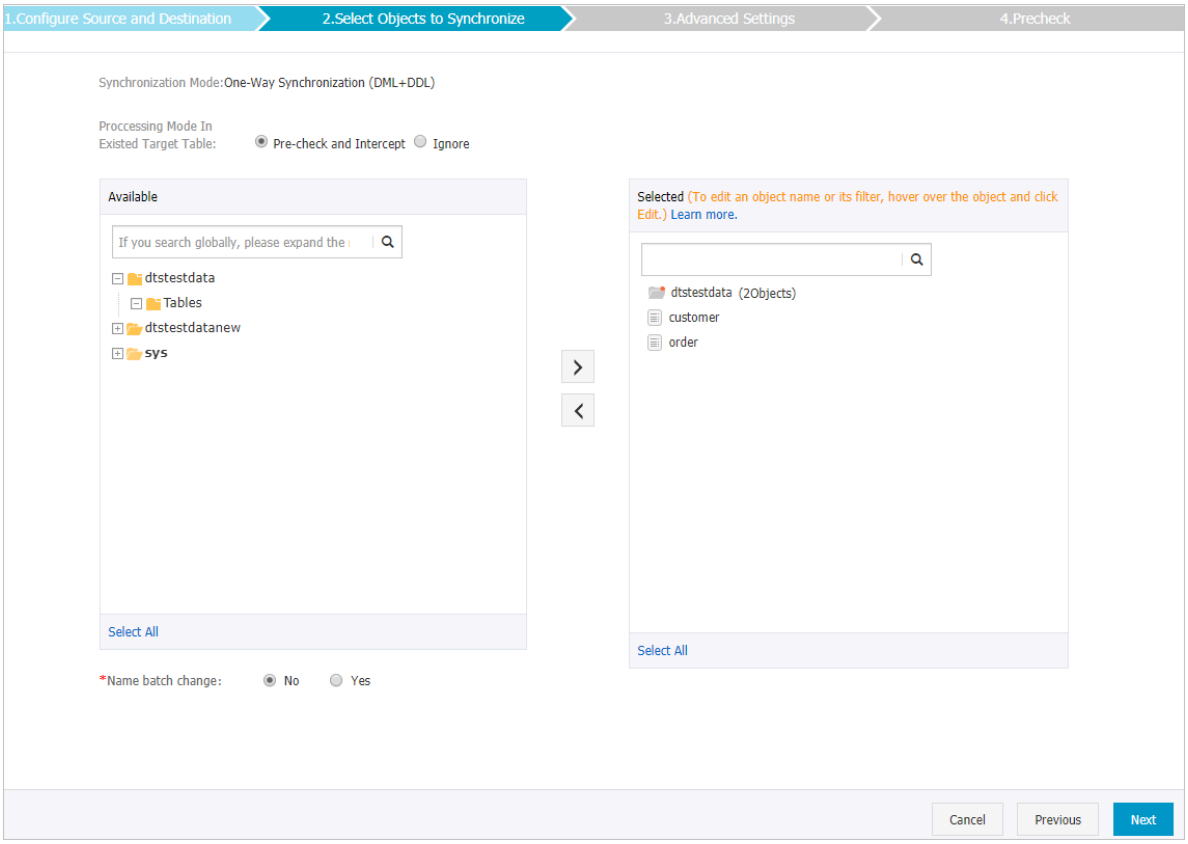
Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.

Section	Parameter	Description
Source Instance Details	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.
	Database Account	Enter the database account for 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. **Note:** The **Encryption** parameter is available only in mainland China and Hong Kong(China).
Destination Instance Details	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	Enter the database account for 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**.

Section	Parameter	Description
	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**.

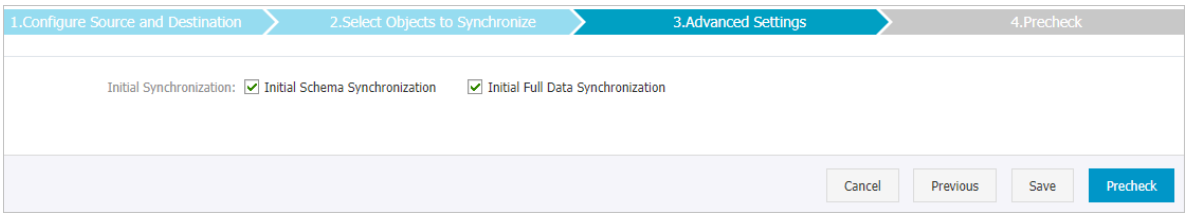
8. Configure the synchronization policy and objects.



Parameter	Description
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.

Parameter	Description
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.



Note:

- 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.

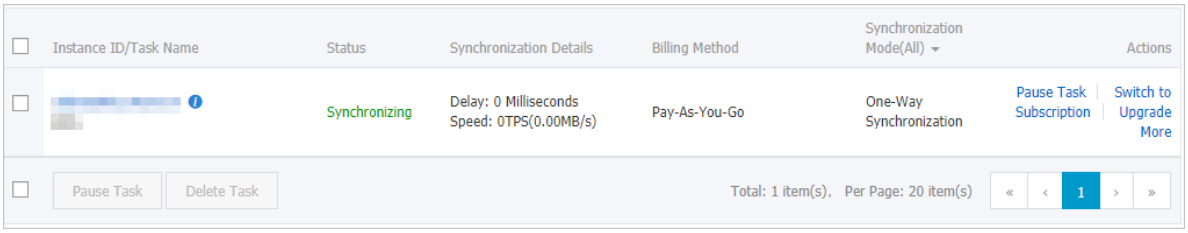
11. In the lower-right corner of the page, click **Precheck**.

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.



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 gh-ost 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

Operation 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

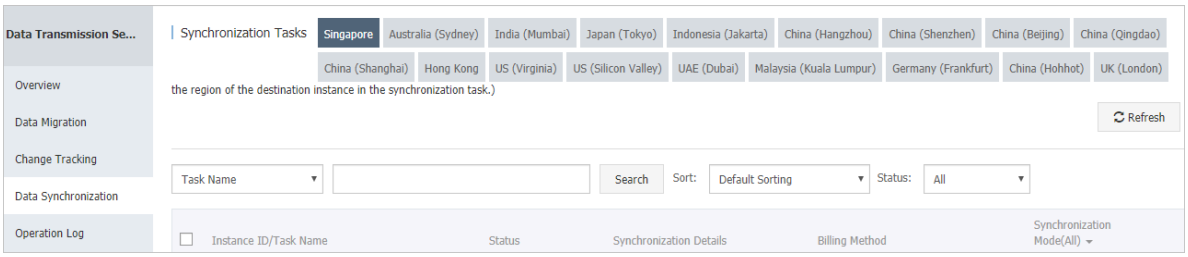
Before configuring the data synchronization task, you must create a database account and configure binary logging. For more information, see #unique_52.

Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

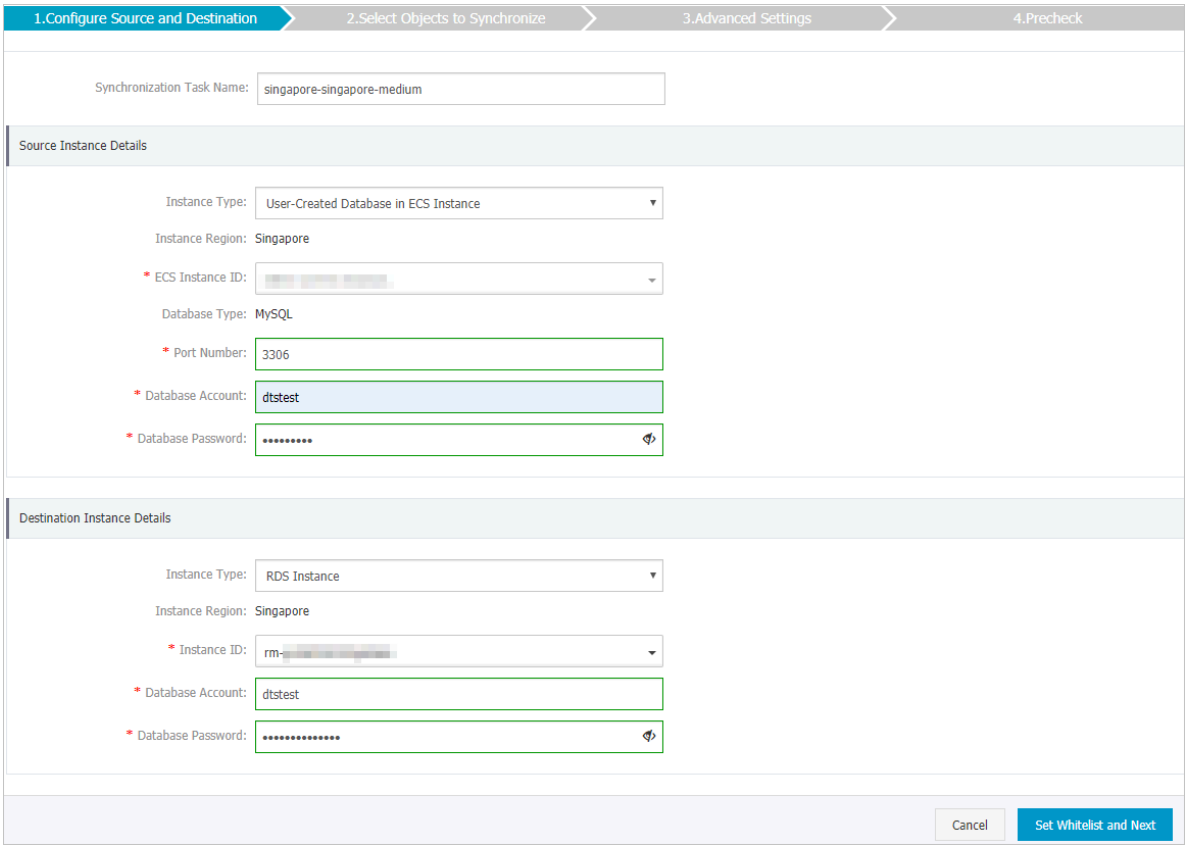
   > **Note:**
   > Select **MySQL** for both the source instance and the destination instance. Select **One-Way Synchronization** as the synchronization topology.

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.

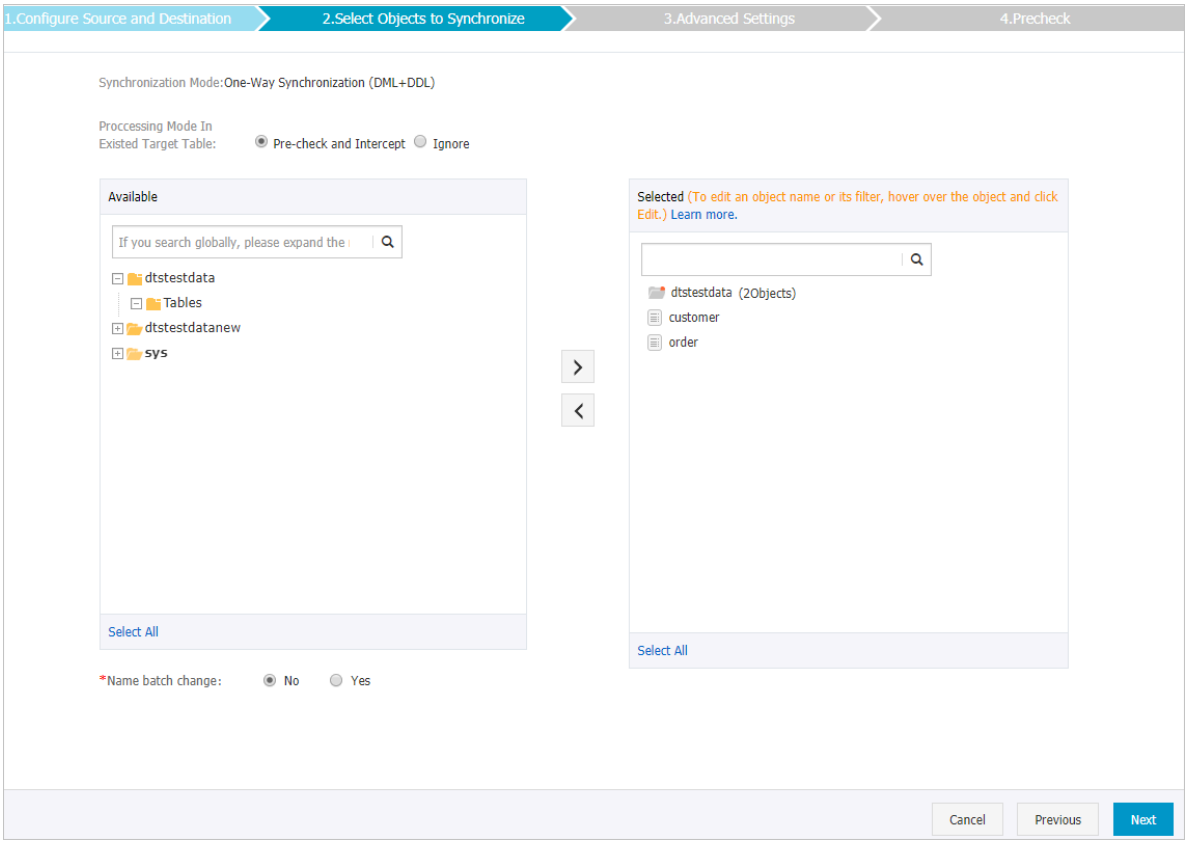


Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **User-Created Database in ECS Instance**.
	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. The default port number is **3306**.

Section	Parameter	Description
	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.
Destination Instance Details	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	Enter the database account for 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**.

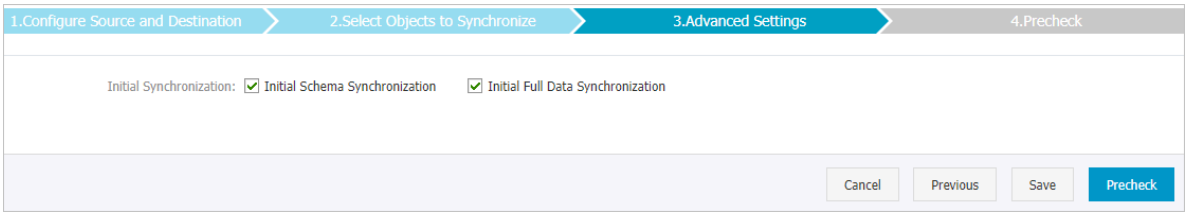
8. Configure the synchronization policy and objects.



Parameter	Description
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.

Parameter	Description
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**.

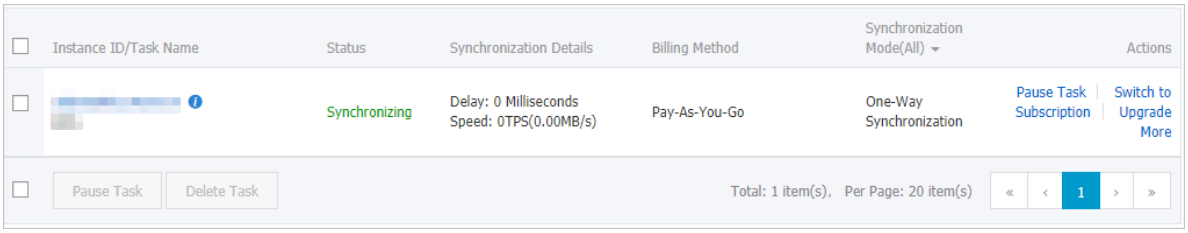
Note:

- 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 synchronization task is in the **Synchronizing** state.

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



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 #unique_53.

Note:
DTS is allowed to access the VPC to which the user-created MySQL database belongs. For more information, see #unique_54.

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 gh-ost 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

Operation 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

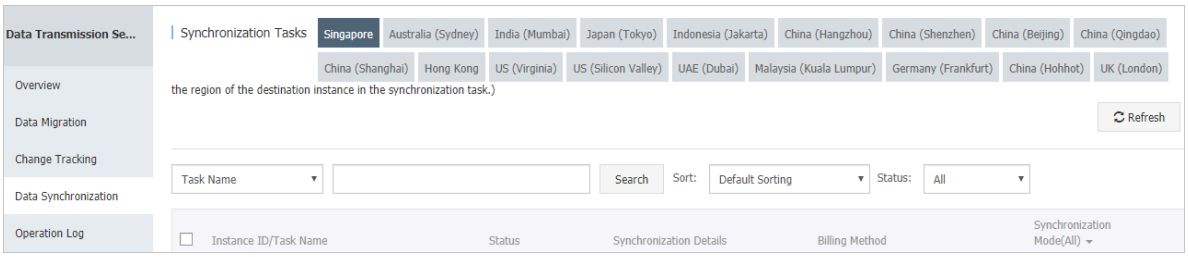
Before configuring the data synchronization task, you must create a database account and configure binary logging. For more information, see #unique_52.

Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

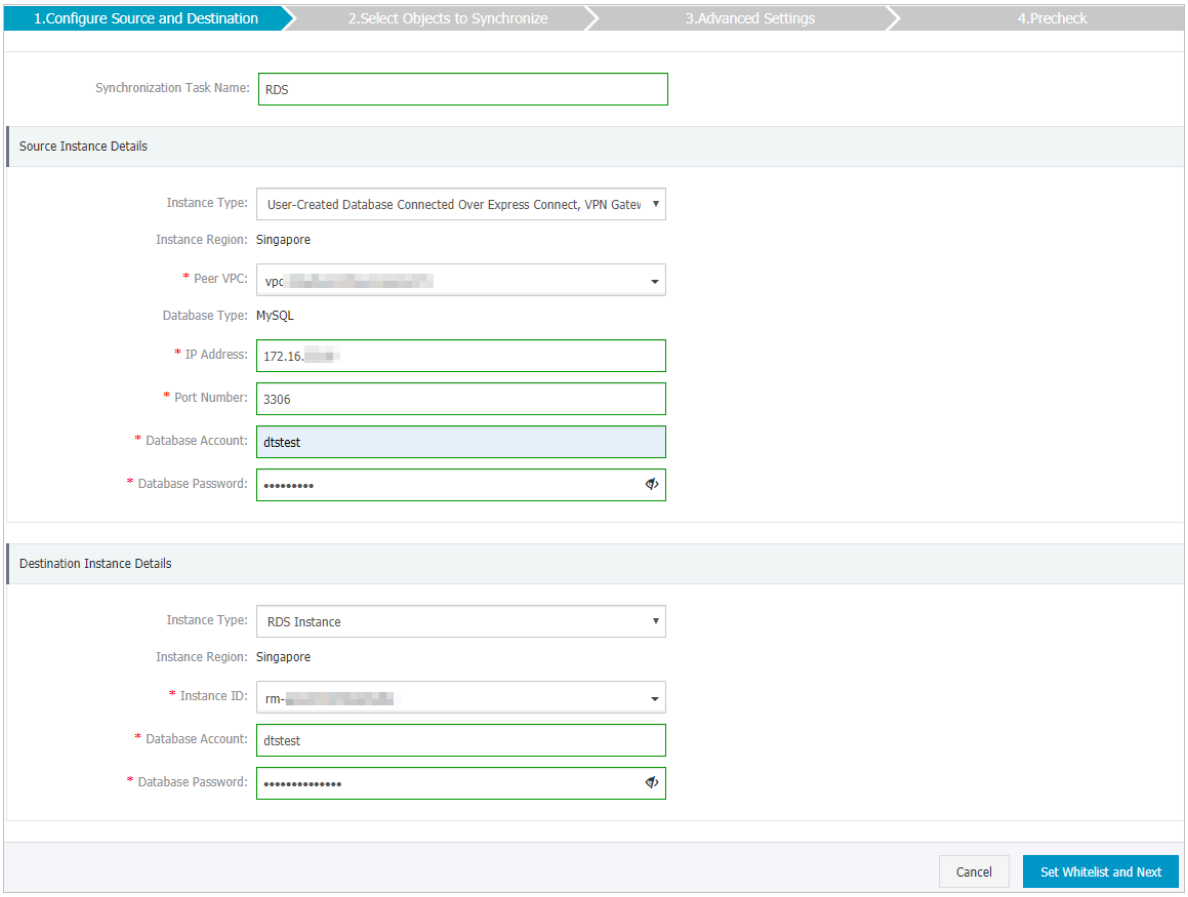
   > **Note:**
   > Select **MySQL** for both the source instance and the destination instance. Select **One-Way Synchronization** as the synchronization topology.

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.

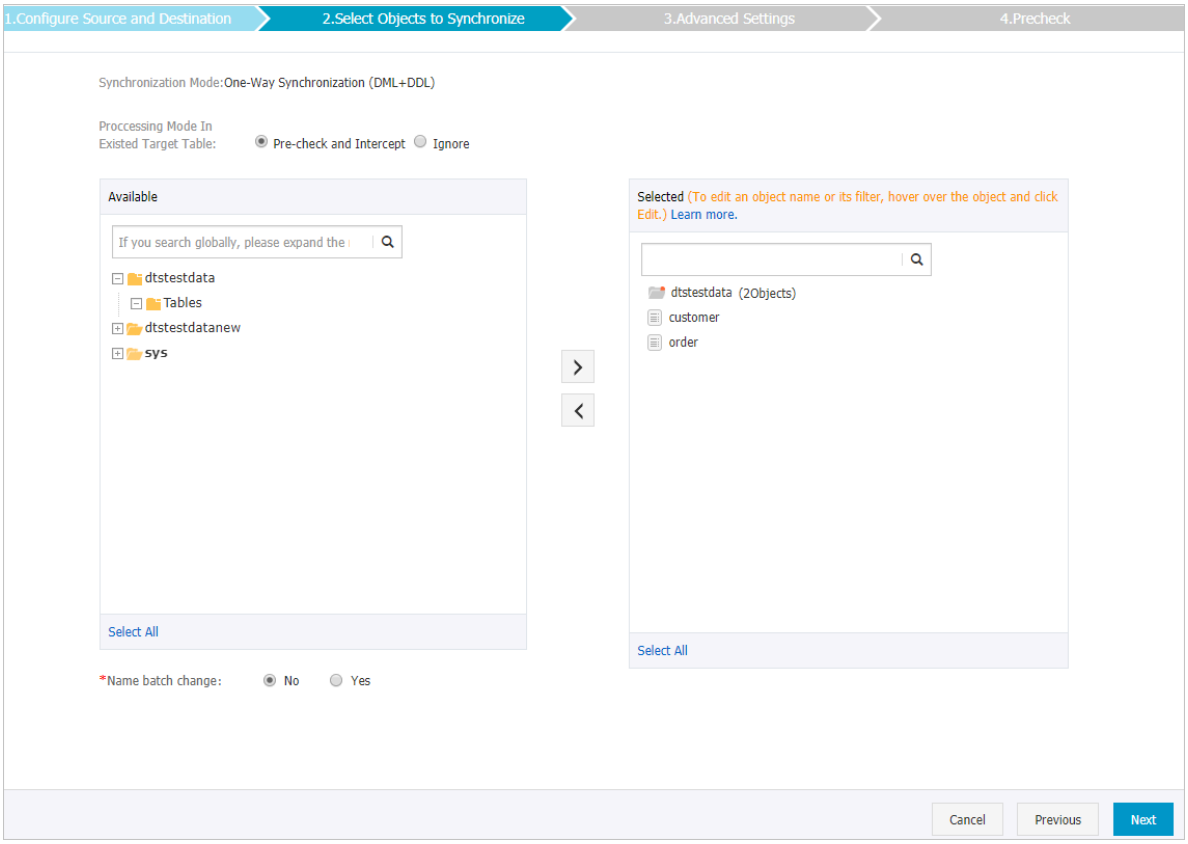


Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **User-Created Database Connected over Express 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.
	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 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.
Destination Instance Details	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	Enter the database account for 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**.

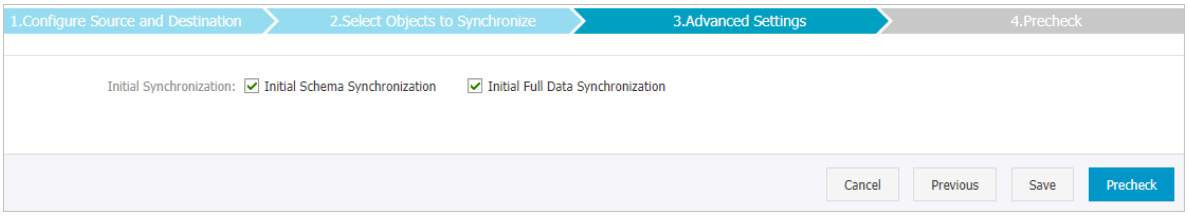
8. Configure the synchronization policy and objects.



Parameter	Description
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.

Parameter	Description
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**.

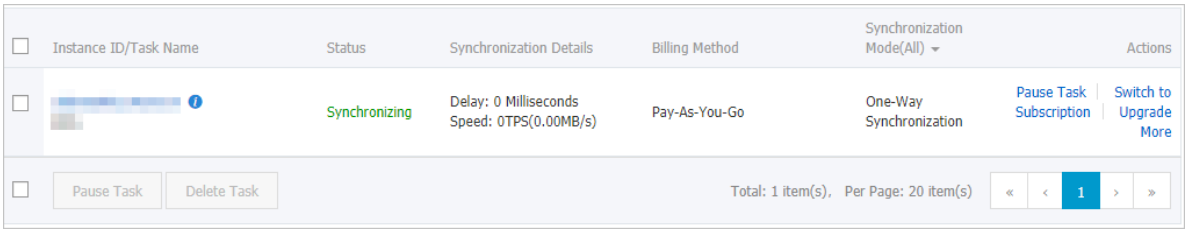
Note:

- 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.



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 #unique_53.

  Note:
  DTS is allowed to access the VPC to which the user-created MySQL database belongs. For more information, see #unique_54.

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 gh-ost 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

Operation 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.

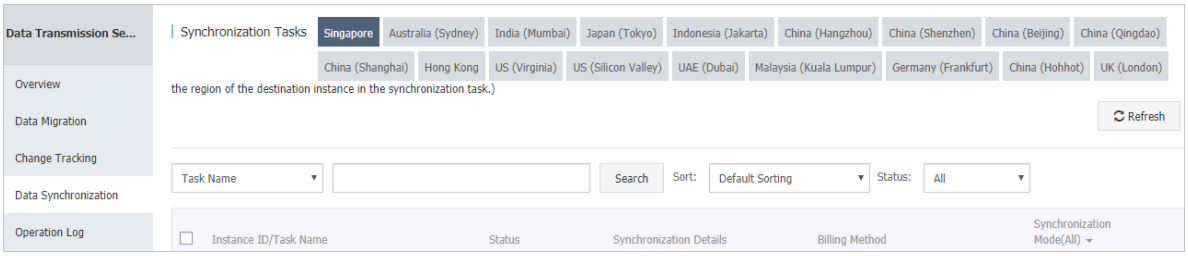
Procedure

1. Create a data synchronization task. For more information, see#unique_51.

   Note:

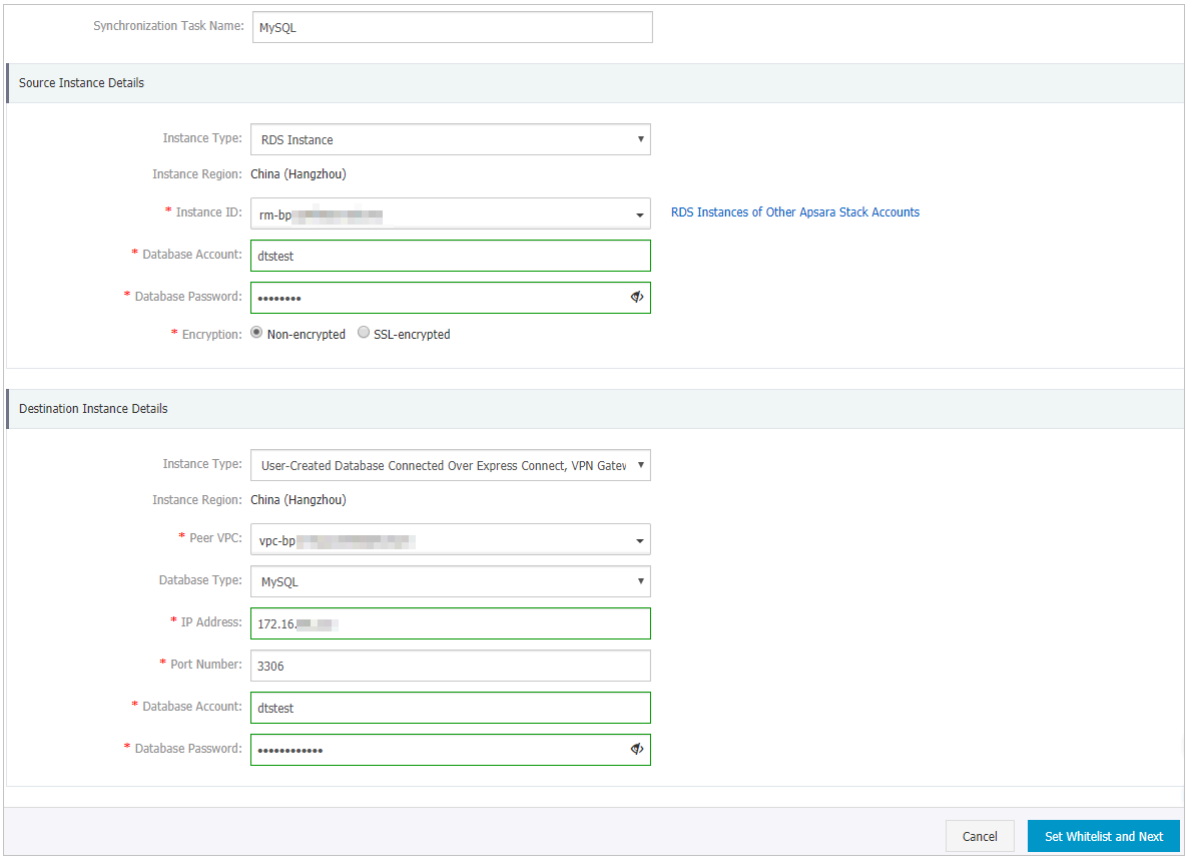
   Select **MySQL** for both the source instance and the destination instance. Select **One-Way Synchronization** as the synchronization topology.

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.

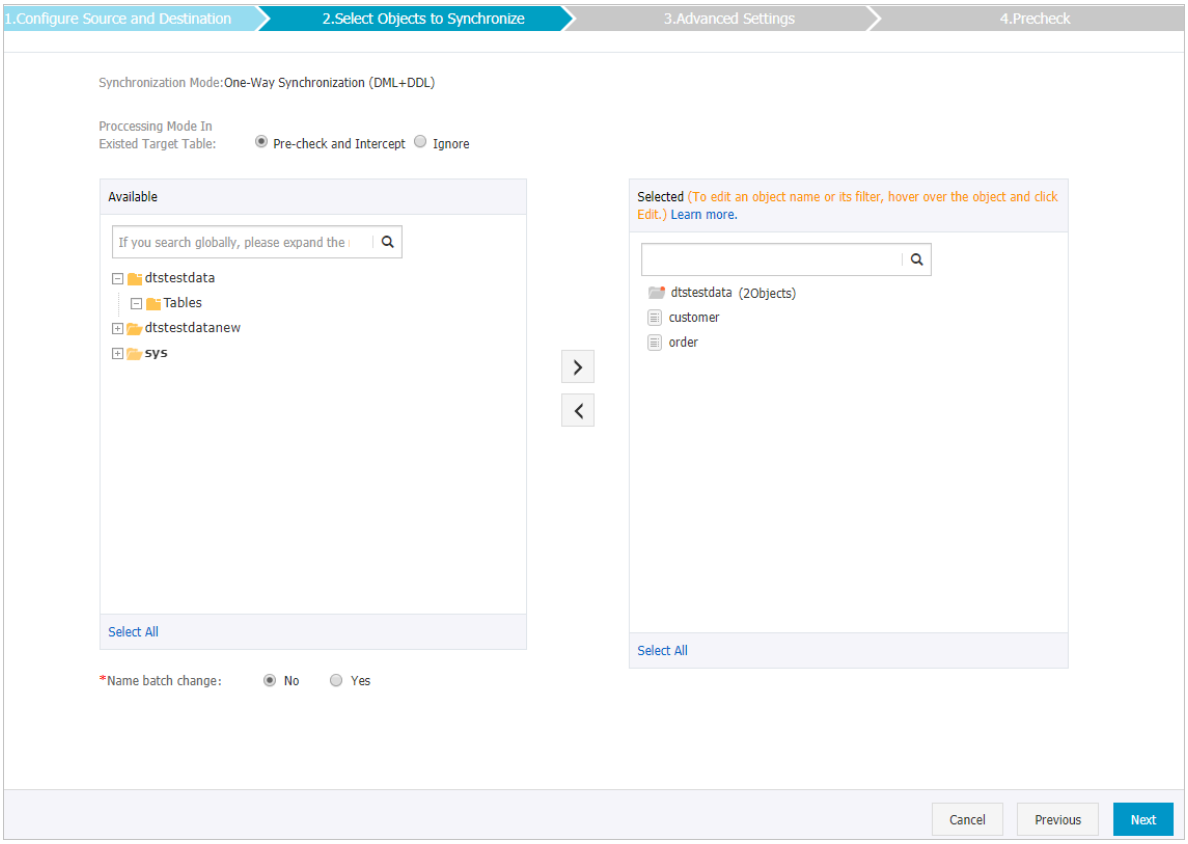


Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **RDS Instance**.
	Instance Region	The region of the source instance that you select when purchasing the data synchronization instance. You cannot change the value of this parameter.
	Instance ID	Select the ID of the source RDS instance.
	Database Account	Enter the database account of the source RDS 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** 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).
Destination Instance Details	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.
	IP Address	Enter the 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. **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.

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

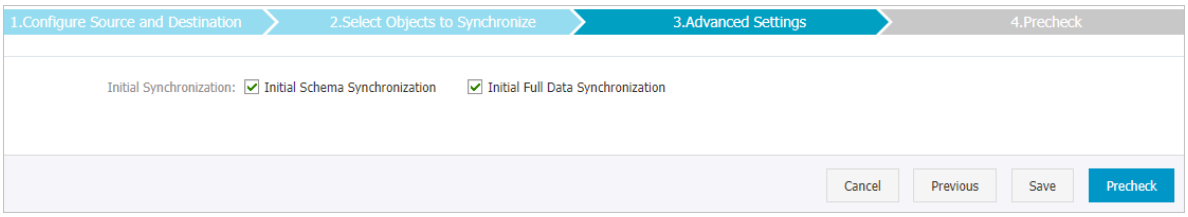
8. Configure the synchronization policy and objects.



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.

Parameter	Description
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.



Note:

- 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.

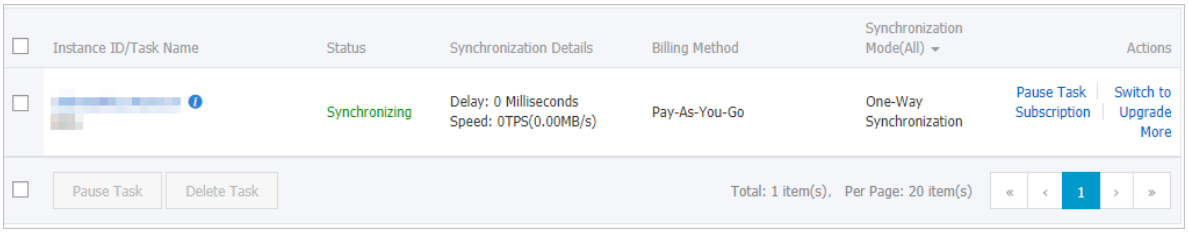
11. In the lower-right corner of the page, click **Precheck**.

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.



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 gh-ost 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

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE

Operation 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

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 #unique_55.

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.

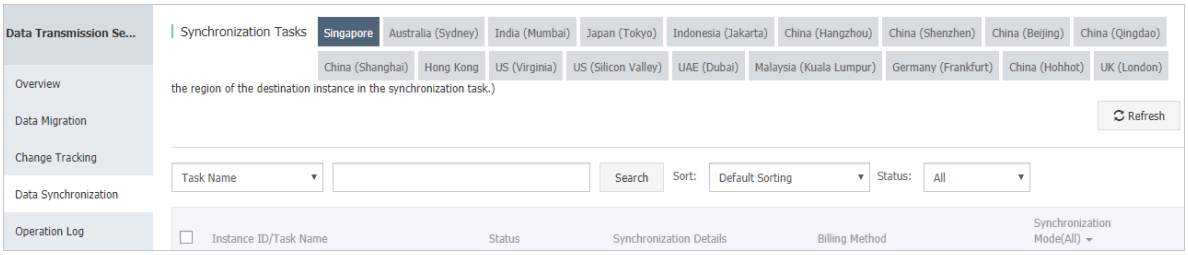
Then, you can create a data migration task or data synchronization task by using the Alibaba Cloud account that owns the destination instance.

Procedure

1. Purchase a data synchronization instance by using the Alibaba Cloud account that owns the destination RDS instance. For more information, see #unique_51/unique_51_Connect_42_section_39h_fto_gdl.

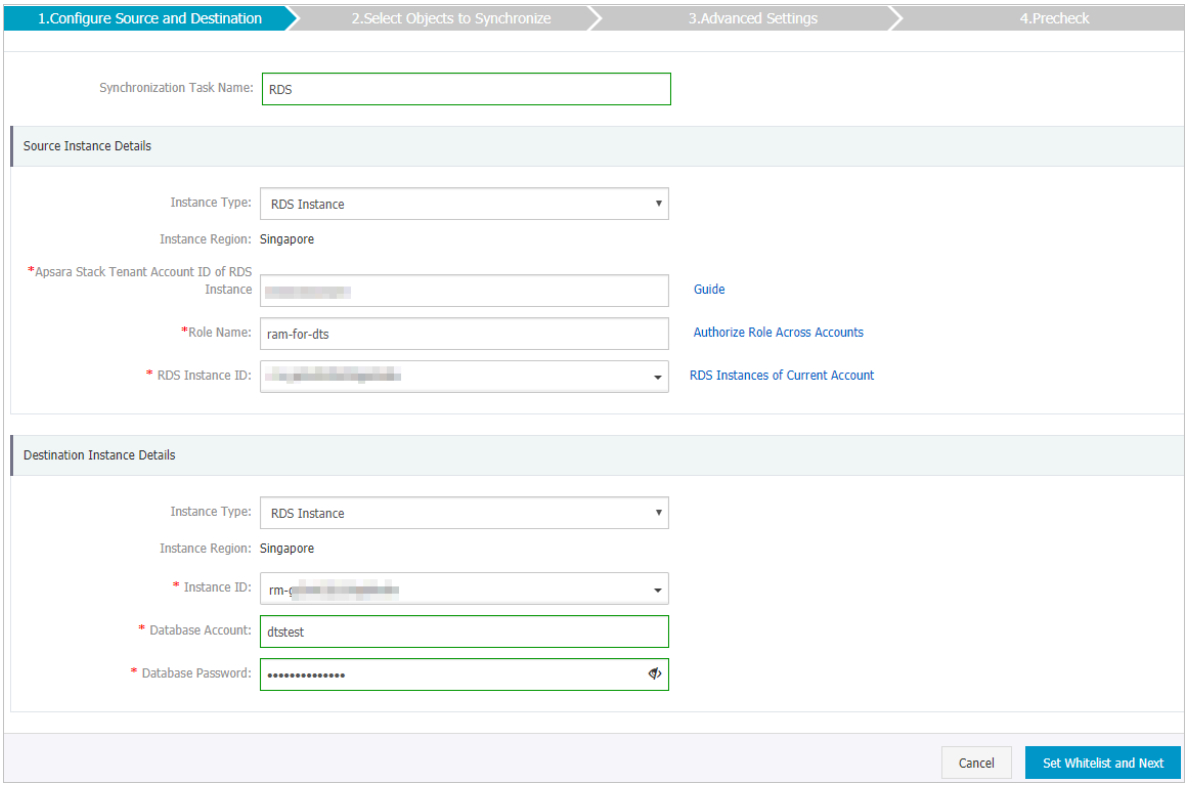
   Note:
   Select **MySQL** for both the source instance and the destination instance. Select **One-Way Synchronization** as the synchronization topology.

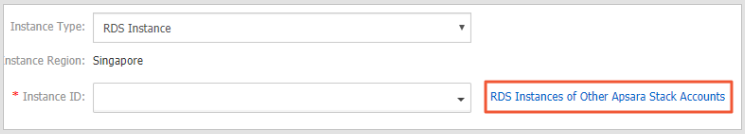
2. Log on to the DTS console with the Alibaba Cloud account that owns the destination RDS instance.
3. In the left-side navigation pane, click **Data Synchronization**.
4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.



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

6. Configure the source and destination instances.

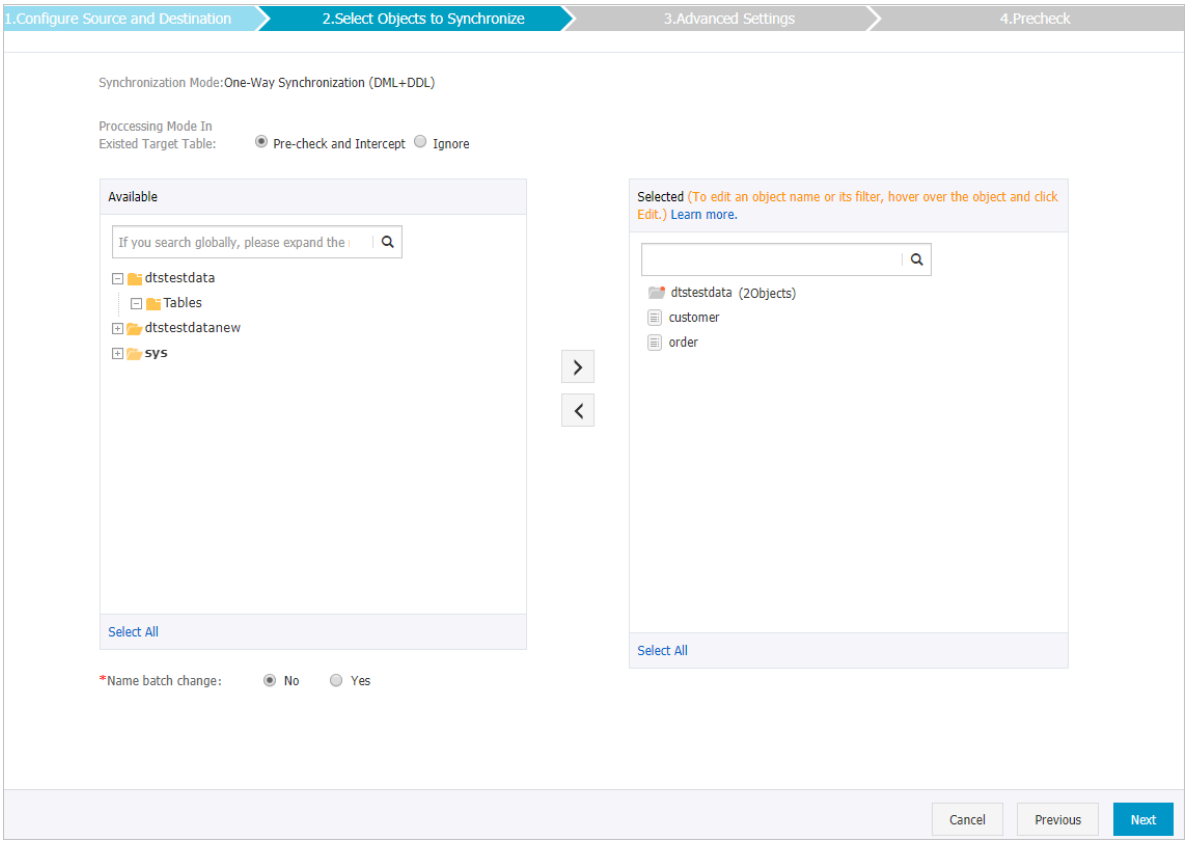


Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **RDS Instance**.
	Instance Region	The region 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.
	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. 

Section	Parameter	Description
	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.
Destination Instance Details	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	Enter the database account for 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**.

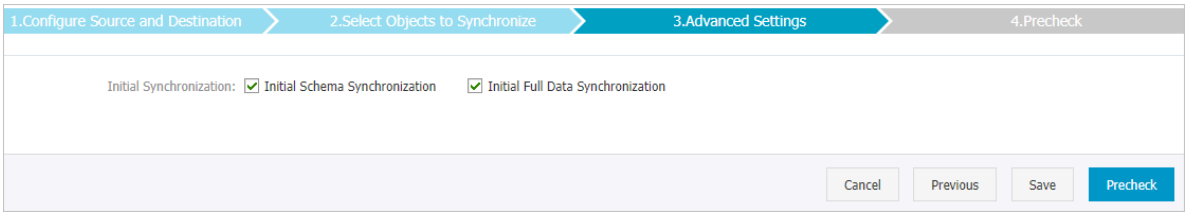
8. Configure the synchronization policy and objects.



Parameter	Description
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.

Parameter	Description
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**.

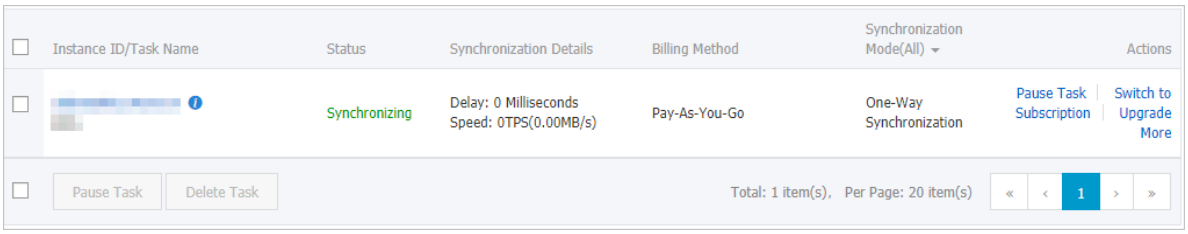
Note:

- 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 gh-ost 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

Operation 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

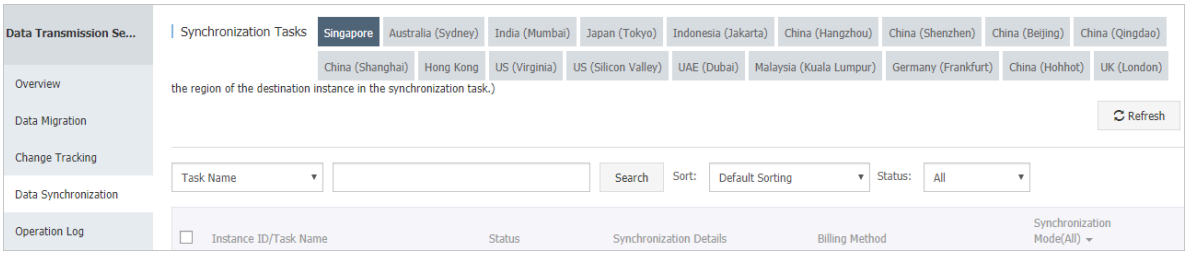
  RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which 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 #unique_51.

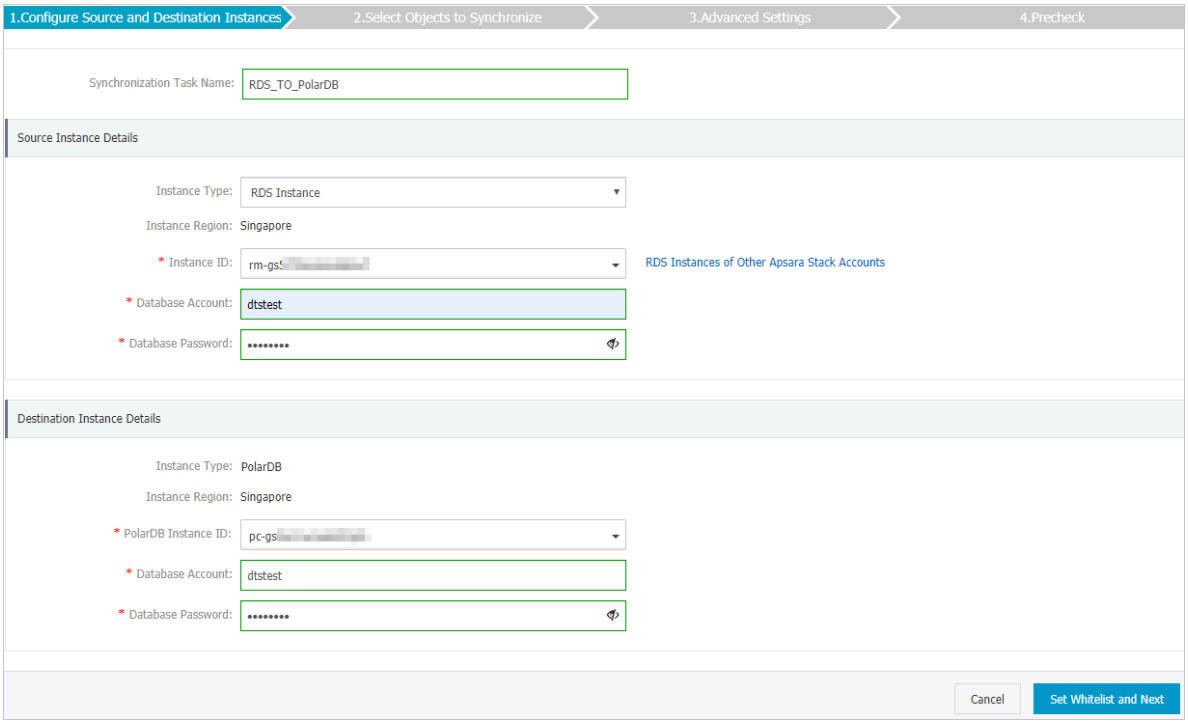
   Note:
   On the buy page, set Source Instance to **MySQL**, Target Instance to **PolarDB**, and Synchronization Topology to **One-Way Synchronization**.

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	Synchronization 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 **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.
	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.

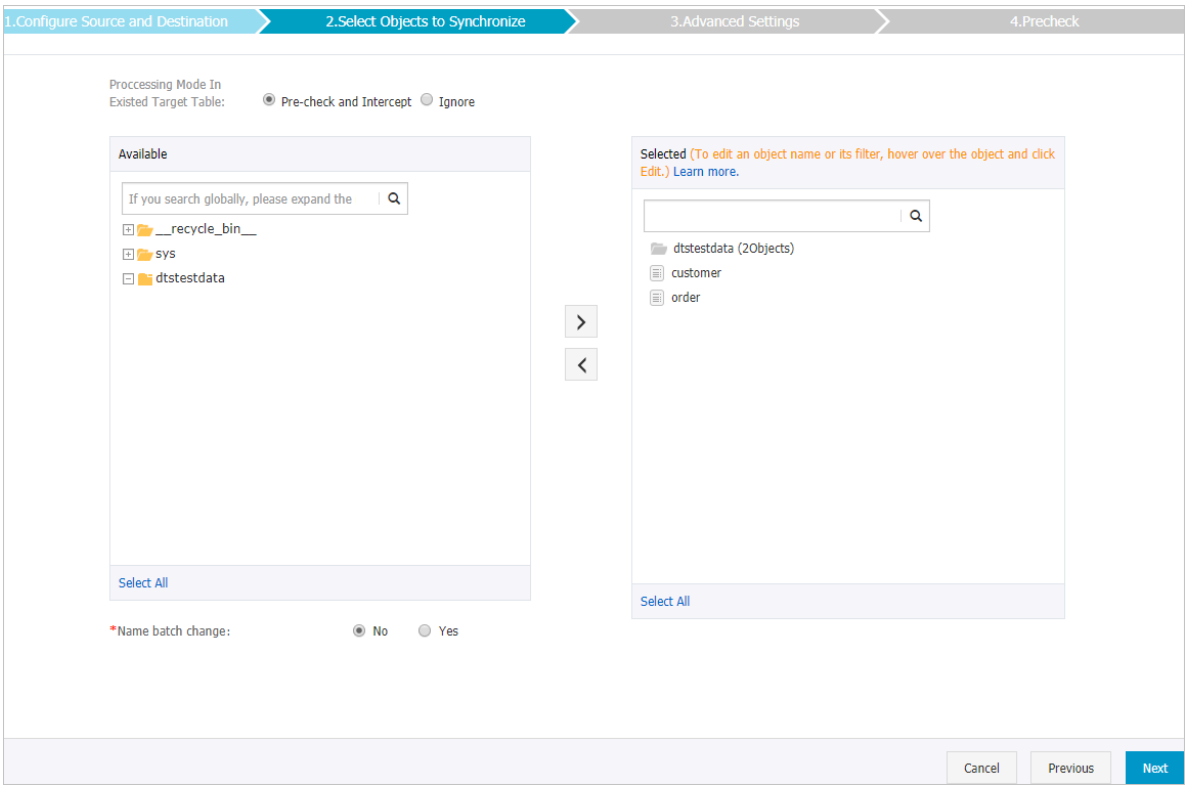
Section	Parameter	Description
	Encryption	Select **Non-encrypted** or **SSL-encrypted**. If you want to select **SSL-encrypted**, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL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

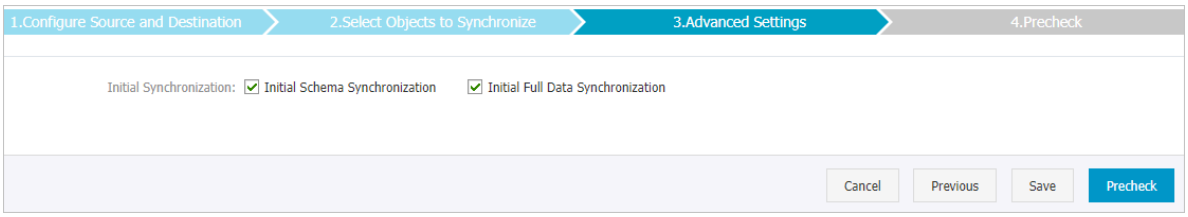
8. Configure the processing mode in existing destination tables and the objects to be synchronized.



Parameter	Description
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. - 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.

Parameter	Description
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.



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.

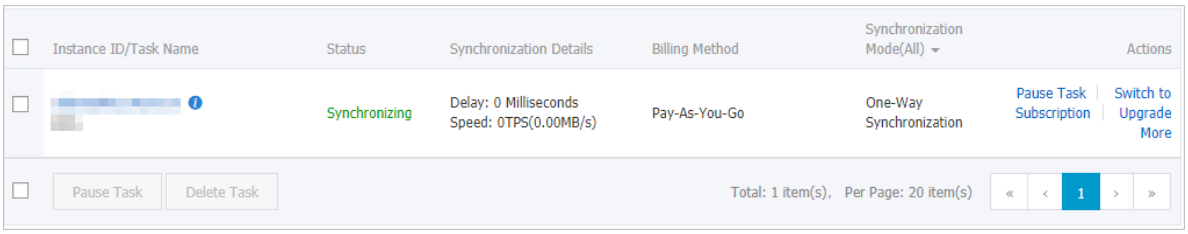
11. In the lower-right corner of the page, click **Precheck**.

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.



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 #unique_57.

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.

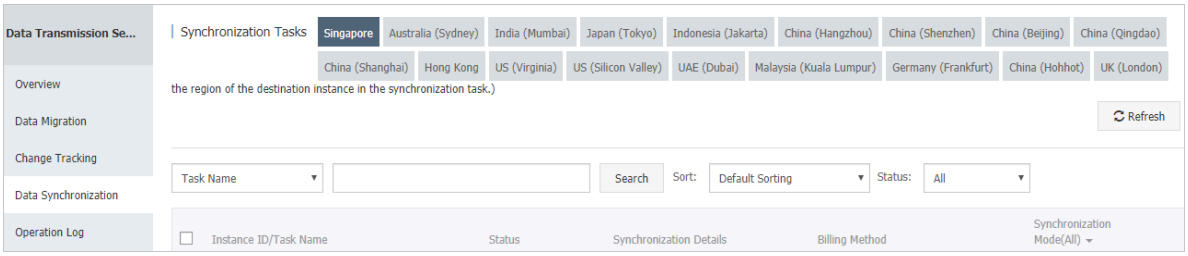
Procedure

1. #unique_51/unique_51_Connect_42_section_39h_fto_gdl.

   > **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**.

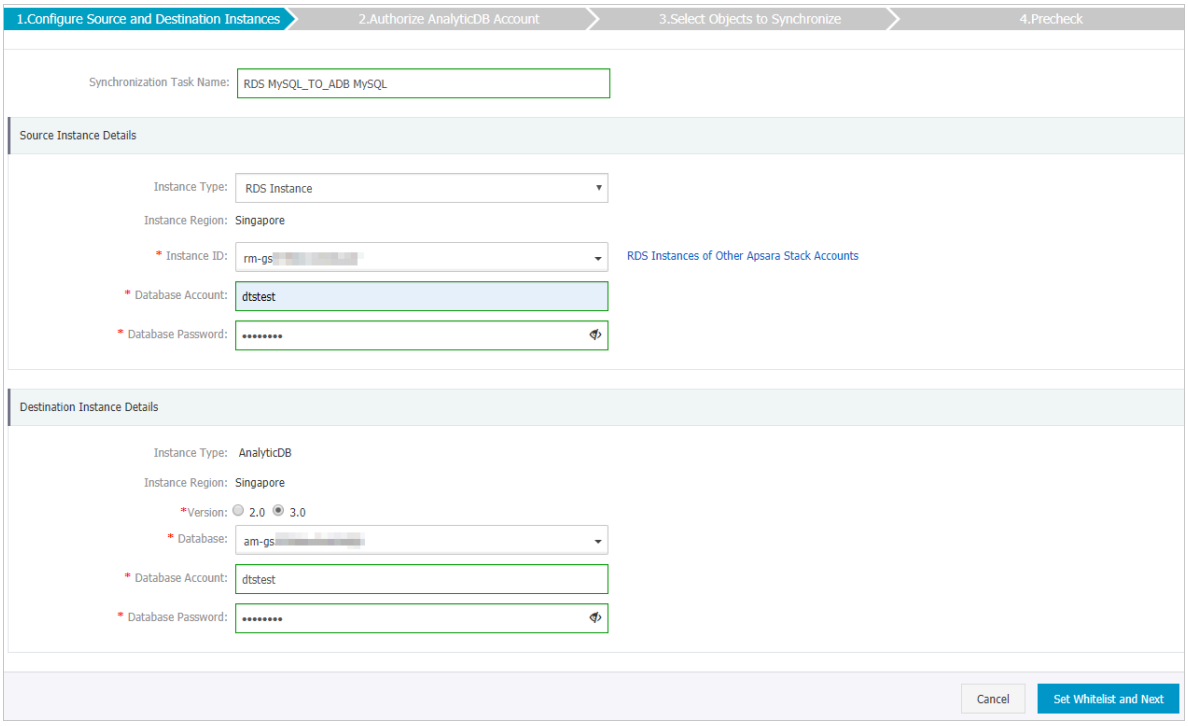
2. Log on to the DTS console.
3. In the left-side navigation pane, click **Data Synchronization**.

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.

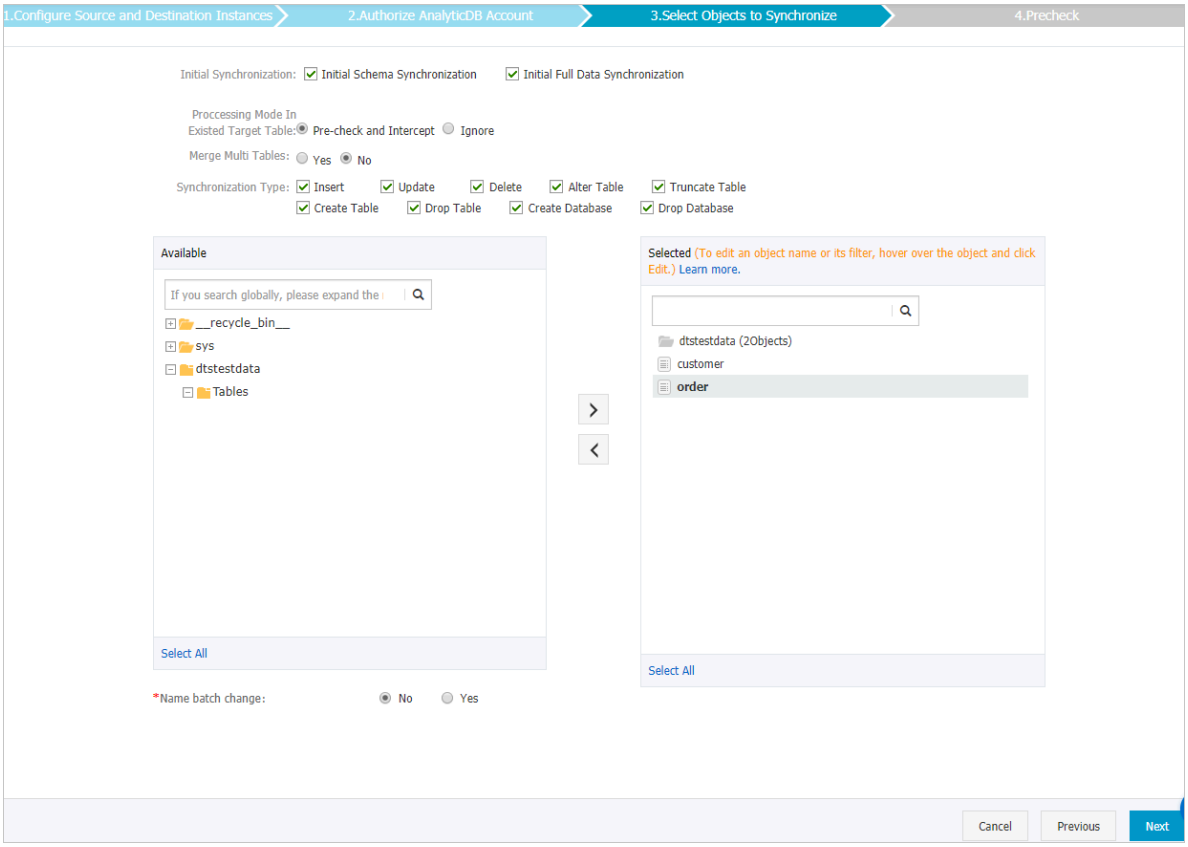


Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **RDS Instance**.
	Instance Region	The 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.

Section	Parameter	Description
	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. **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 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.
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 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.

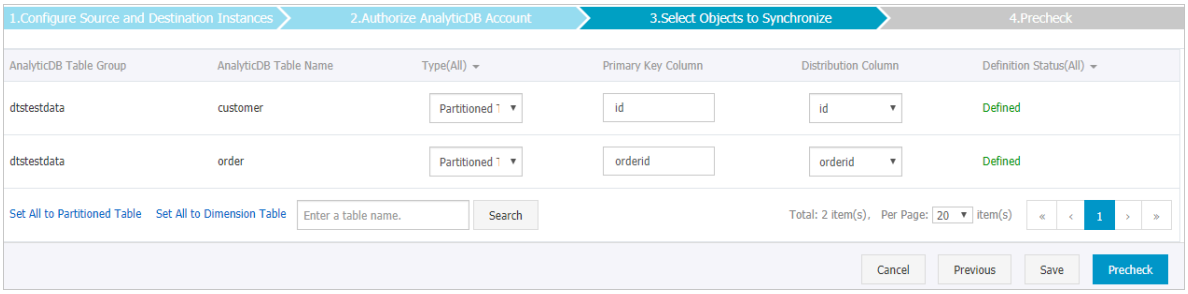


Parameter	Description
Initial Synchronization	You must select both **Initial Schema Synchronization** and **Initial Full Data Synchronization** in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination cluster. The schemas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

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 the `__dts_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.

Parameter	Description
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.
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. • If you select a table as the object to be synchronized, only ADD COLUMN operations on the 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 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. 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.

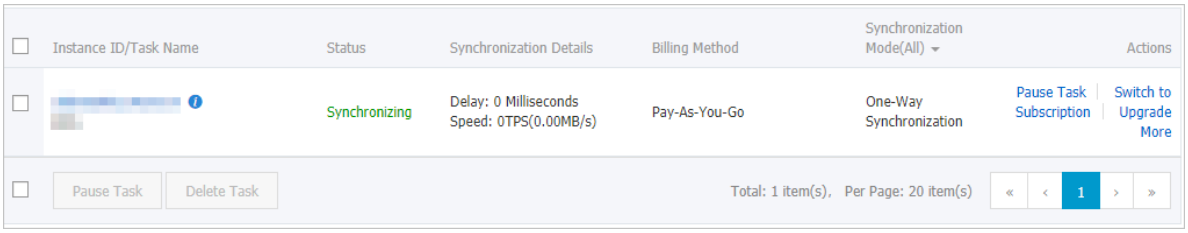
11. In the lower-right corner of the page, click **Precheck**.

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

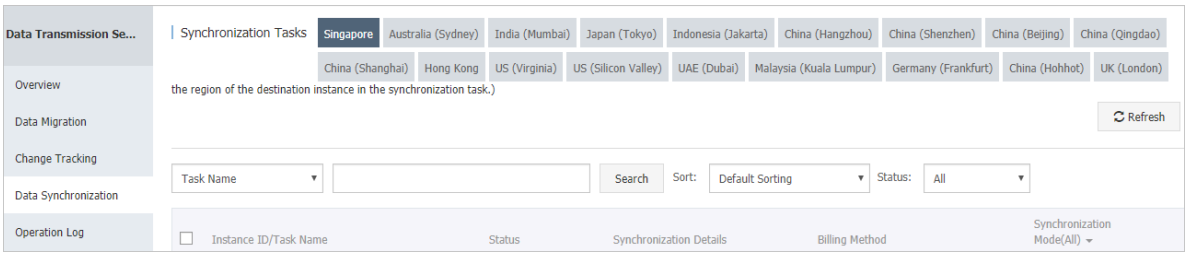
Term in MySQL	Term in AnalyticDB for PostgreSQL
Database	Schema
Table	Table

Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

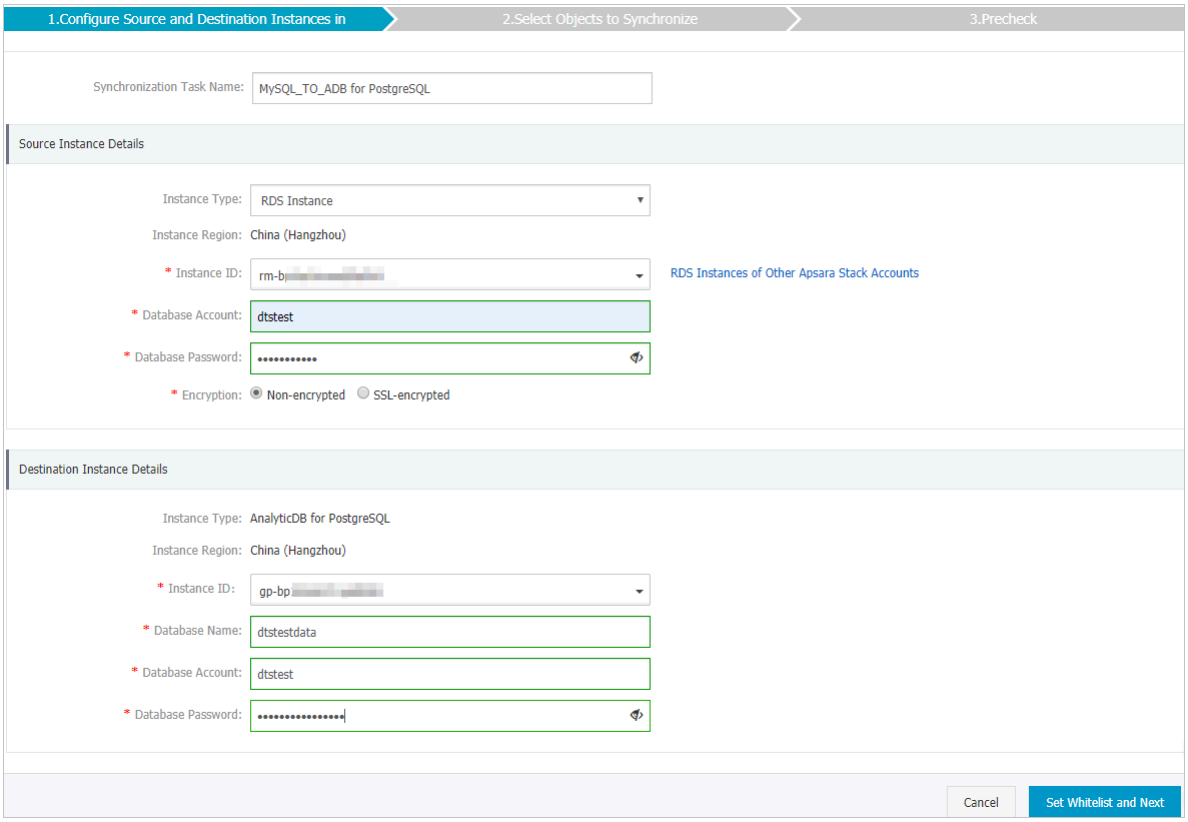
> **Note:**
> On the purchase page, select **MySQL** for the source instance and **AnalyticDB for PostgreSQL** for the destination instance. Select **One-Way Synchronization** as the synchronization topology.

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	Synchronization 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 **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.
	Instance ID	Select the ID of the source RDS instance.
	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**.

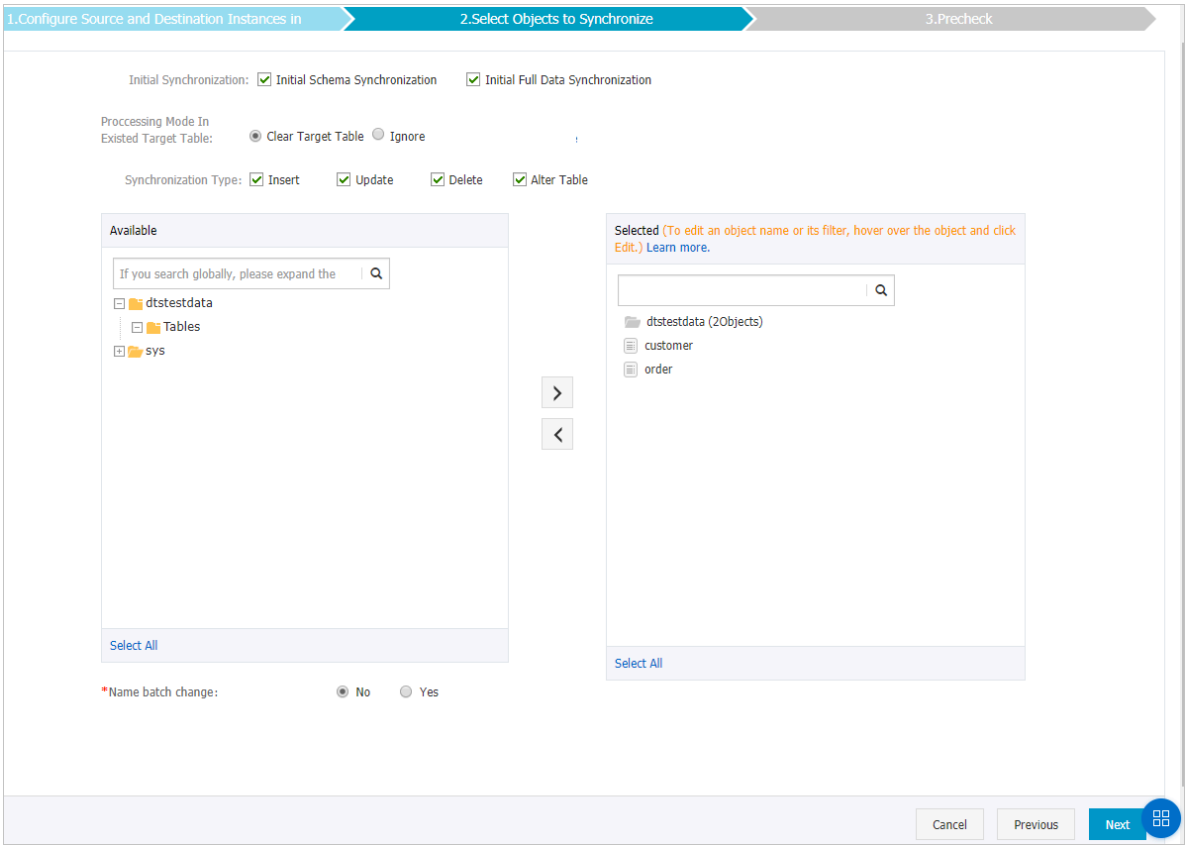
Section	Parameter	Description
	Database Password	Enter the password for the database account.
	Encryption	Select **Non-encrypted** or **SSL-encrypted**. If you want to select **SSL-encrypted**, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more 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).
Destination Instance Details	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.
	Database Name	The name of the destination database.
	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.

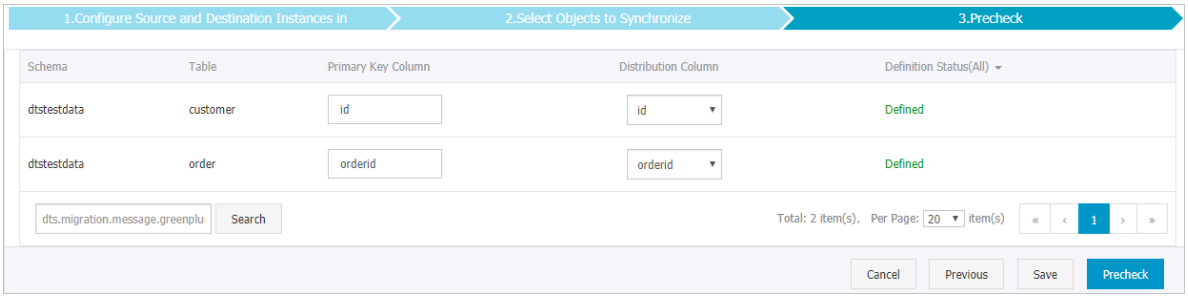


Section	Parameter	Description
Synchronization policy	Initial Synchronization	You must select both **Initial Schema Synchronization** and **Initial Full Data Synchronization** in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the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

Section	Parameter	Description
	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.
	Synchronization Type	Select the types of operations that you want to synchronize based on your business requirements. • **Insert** • **Update** • **Delete** • **Alter Table**

Section	Parameter	Description
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.



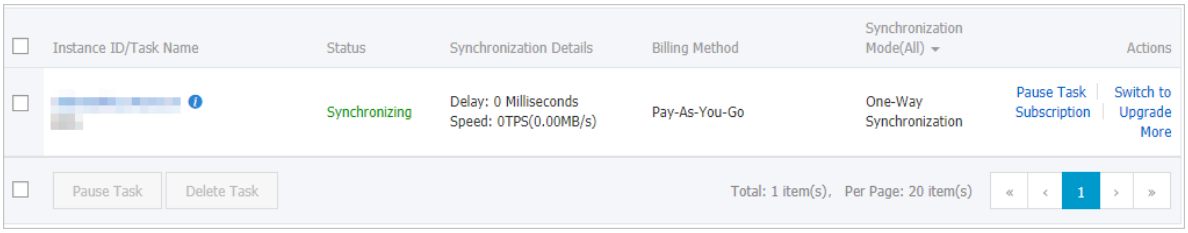
Note:
The page in this step appears only if you select **Initial Schema Synchronization**.

10. 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.

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 #unique_52.

  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 performanc e 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

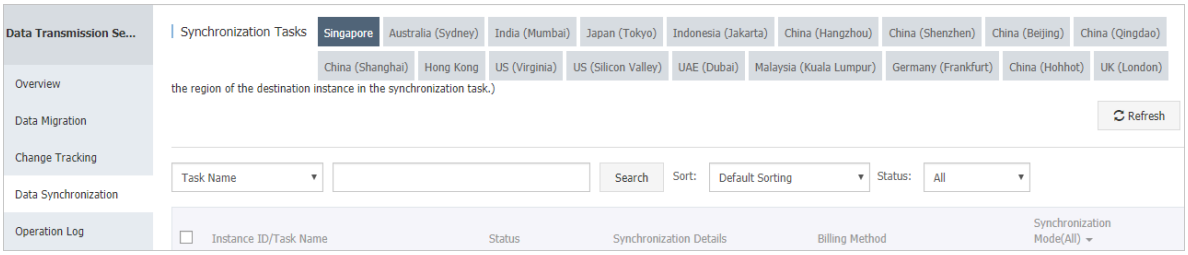
Term in MySQL	Term in AnalyticDB for PostgreSQL
Database	Schema
Table	Table

Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

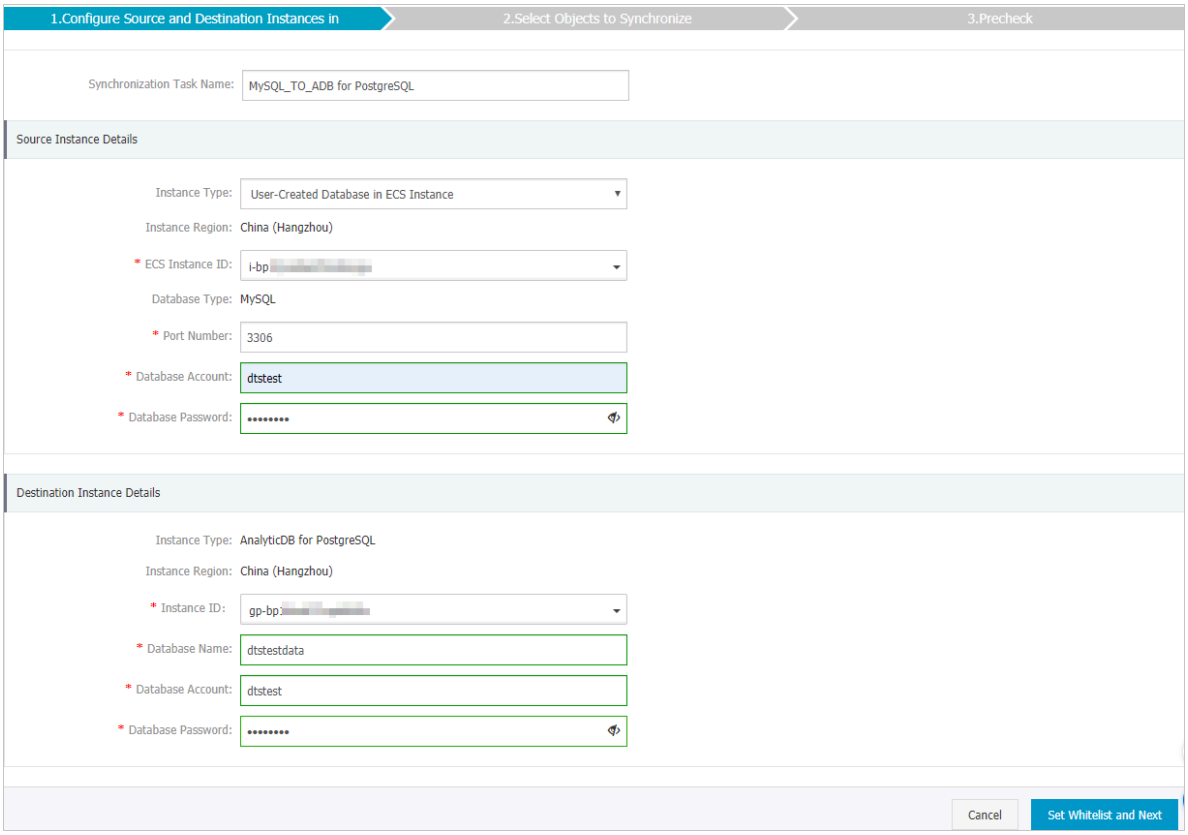
   > **Note:**
   > On the purchase page, select **MySQL** for the source instance and **AnalyticDB for PostgreSQL** for the destination instance. Select **One-Way Synchronization** as the synchronization topology.

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	Synchronization 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**.
	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**.

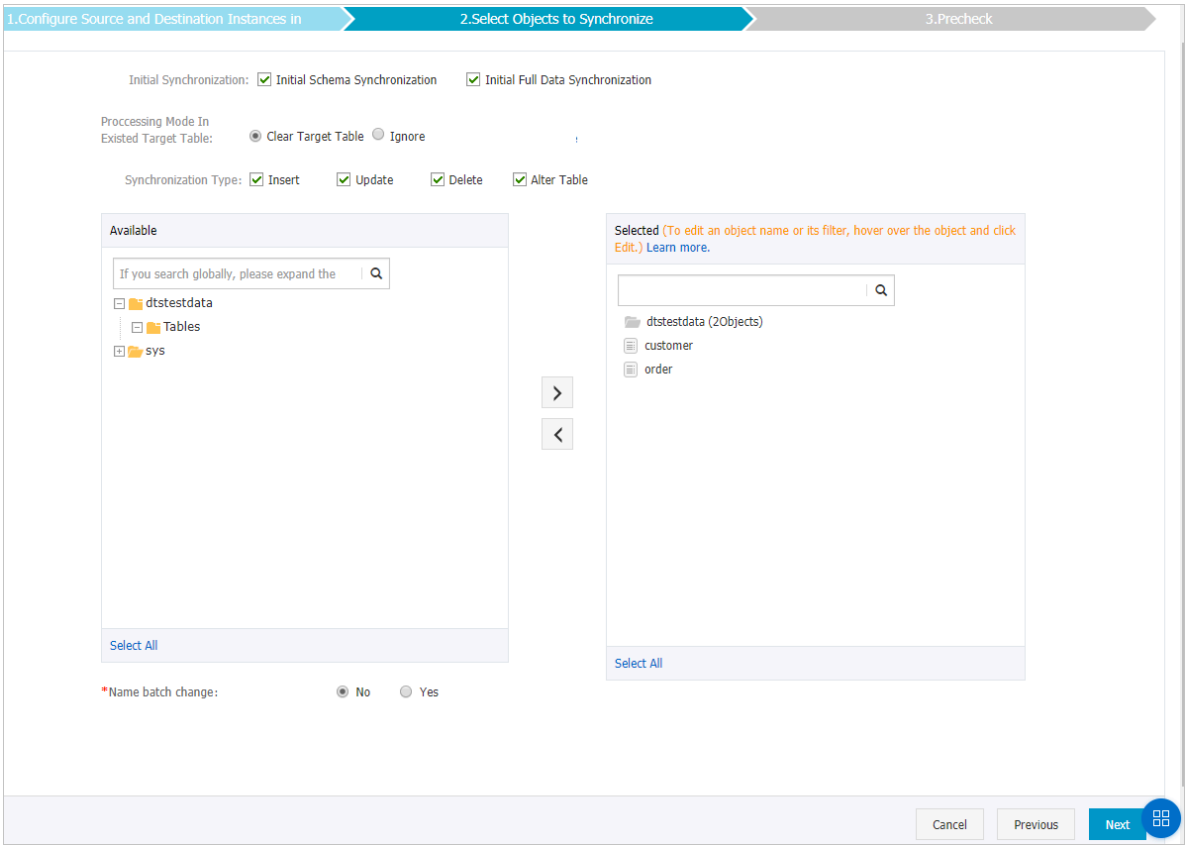
Section	Parameter	Description
	Database Account	Enter the account for the user-created MySQL database. **Note:** The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.
	Database Password	Enter the password for the database account.
Destination Instance Details	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.
	Database Name	Enter the name of the destination database.
	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 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.

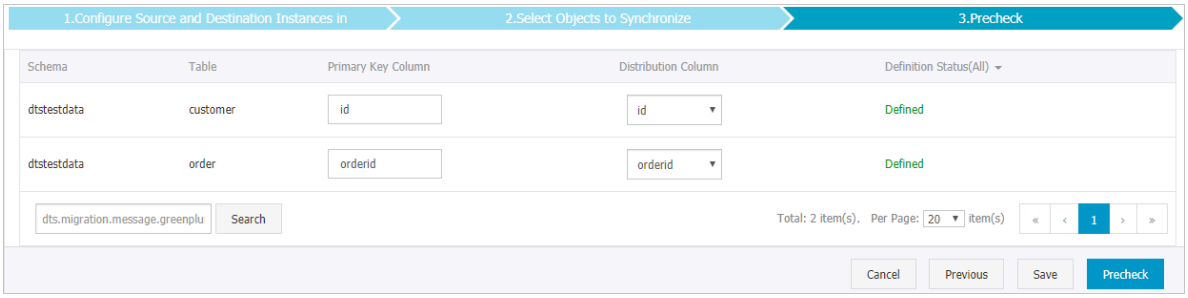


Section	Parameter	Description
Synchronization policy	Initial Synchronization	You must select both **Initial Schema Synchronization** and **Initial Full Data Synchronization** in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the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

Section	Parameter	Description
	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.
	Synchronization Type	Select the types of operations that you want to synchronize based on your business requirements. • **Insert** • **Update** • **Delete** • **Alter Table**

Section	Parameter	Description
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.



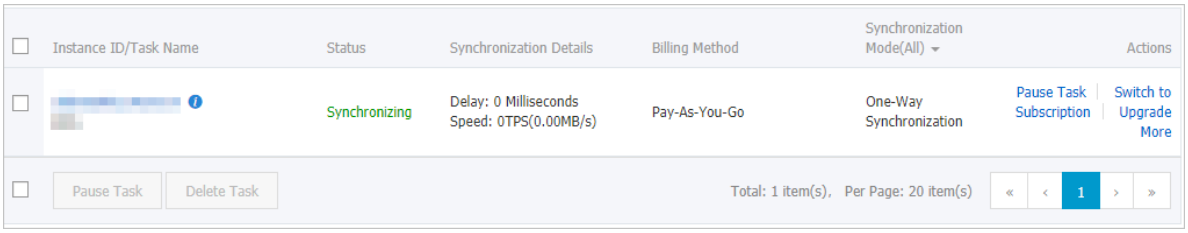
Note:
The page in this step appears only if you select **Initial Schema Synchronization**.

10. 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.

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 #unique_52.

  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 #unique_54.

  Note:

For more information about how to connect a VPC to an on-premises data center, see #unique_53.

- 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

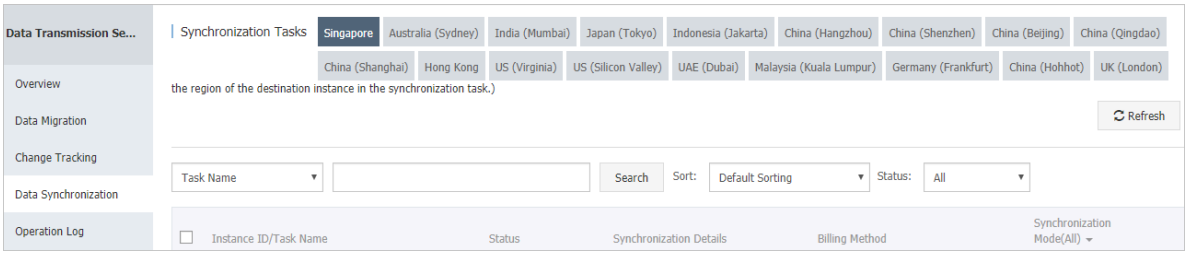
Term in MySQL	Term in AnalyticDB for PostgreSQL
Database	Schema
Table	Table

Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

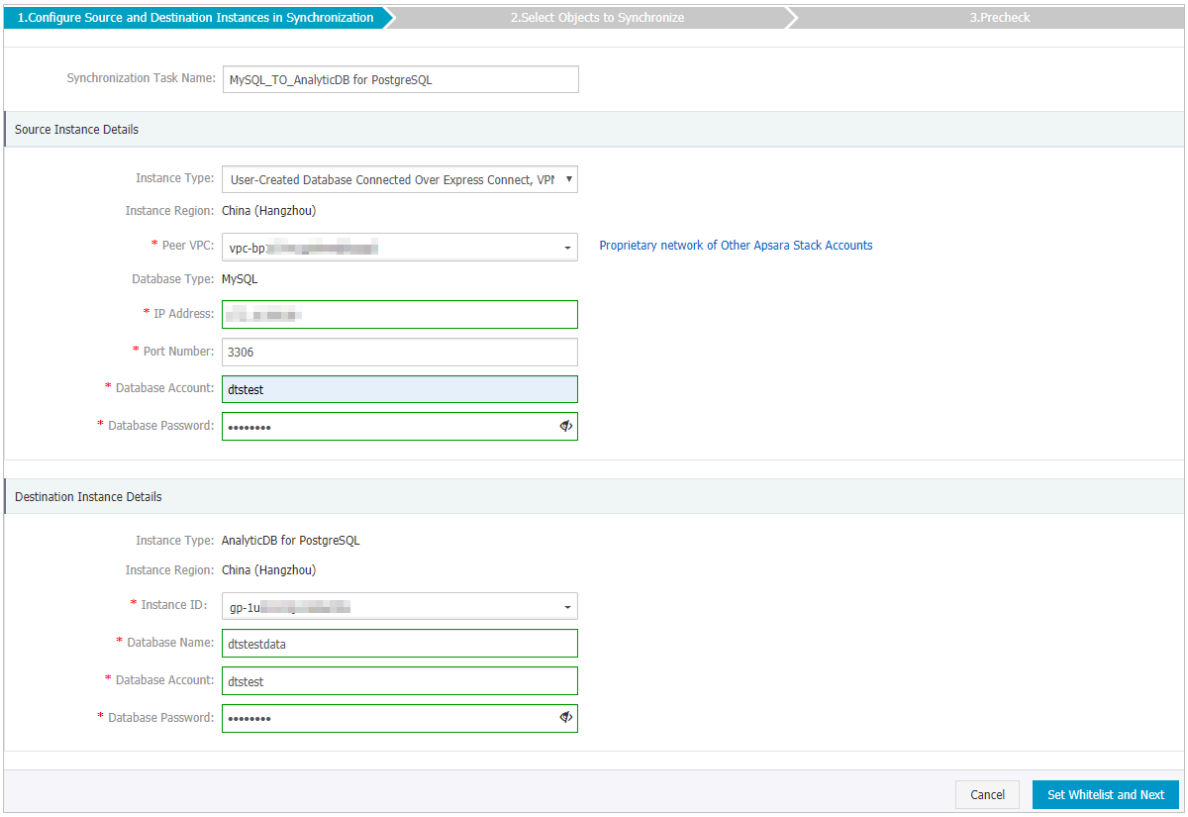
   Note:
   On the buy page, set Source Instance to **MySQL**, set Target Instance to **AnalyticDB for PostgreSQL**, and set Synchronization Topology to **One-Way Synchronization**.

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.



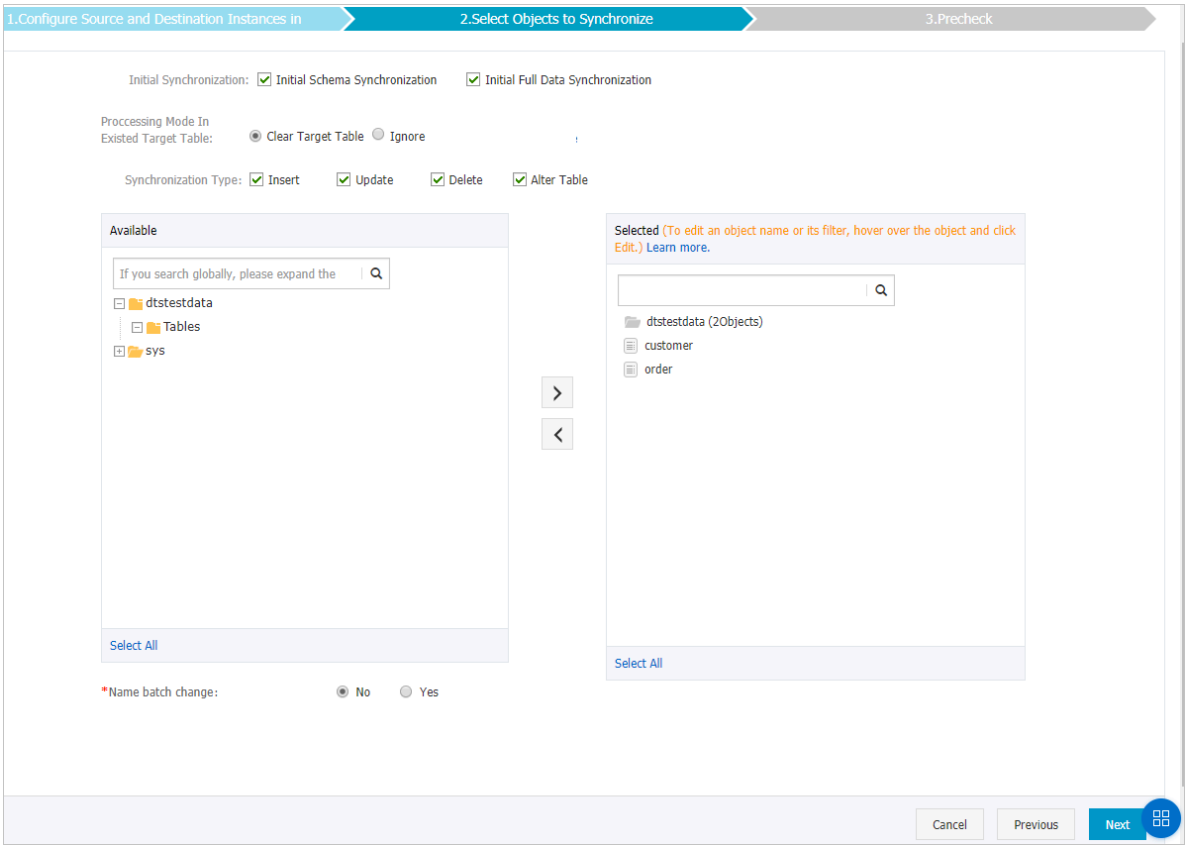
Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **User-Created Database Connected over 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.
	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. **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.
Destination Instance Details	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.
	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.

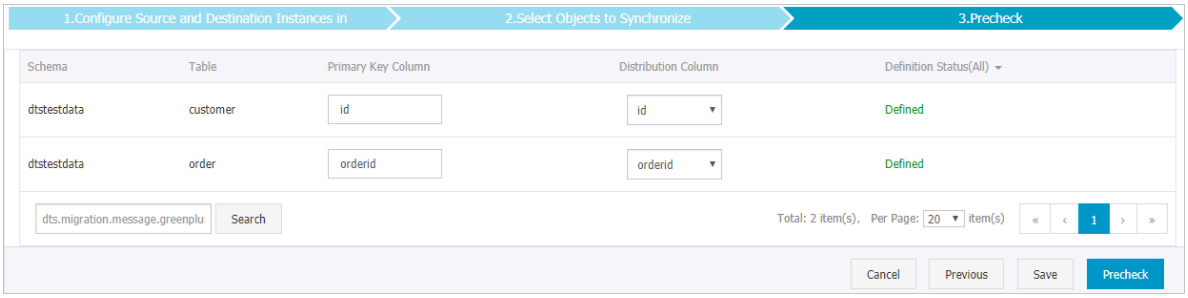


Section	Parameter	Description
Synchronization policy	Initial Synchronization	You must select both **Initial Schema Synchronization** and **Initial Full Data Synchronization** in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the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

Section	Parameter	Description
	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 requiremen ts. • **Insert** • **Update** • **Delete** • **Alter Table**

Section	Parameter	Description
Objects to be synchronized	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. 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**.

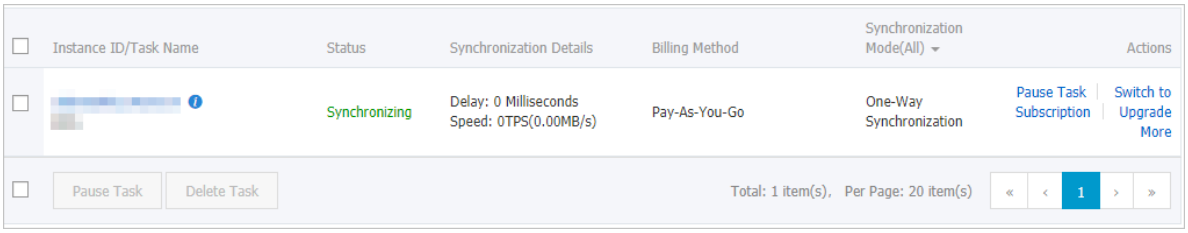
Note:

- 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.

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 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

#unique_52

Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

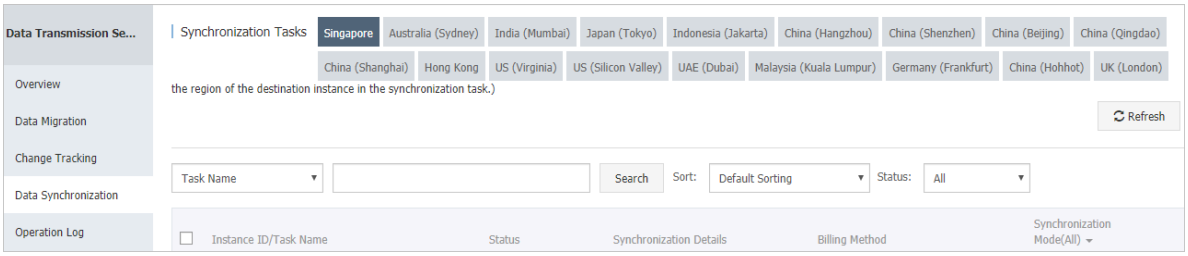
   Note:

   Select **MySQL** for the source instance and **Elasticsearch** for the destination instance.

   Select **One-Way Synchronization** as the synchronization topology.

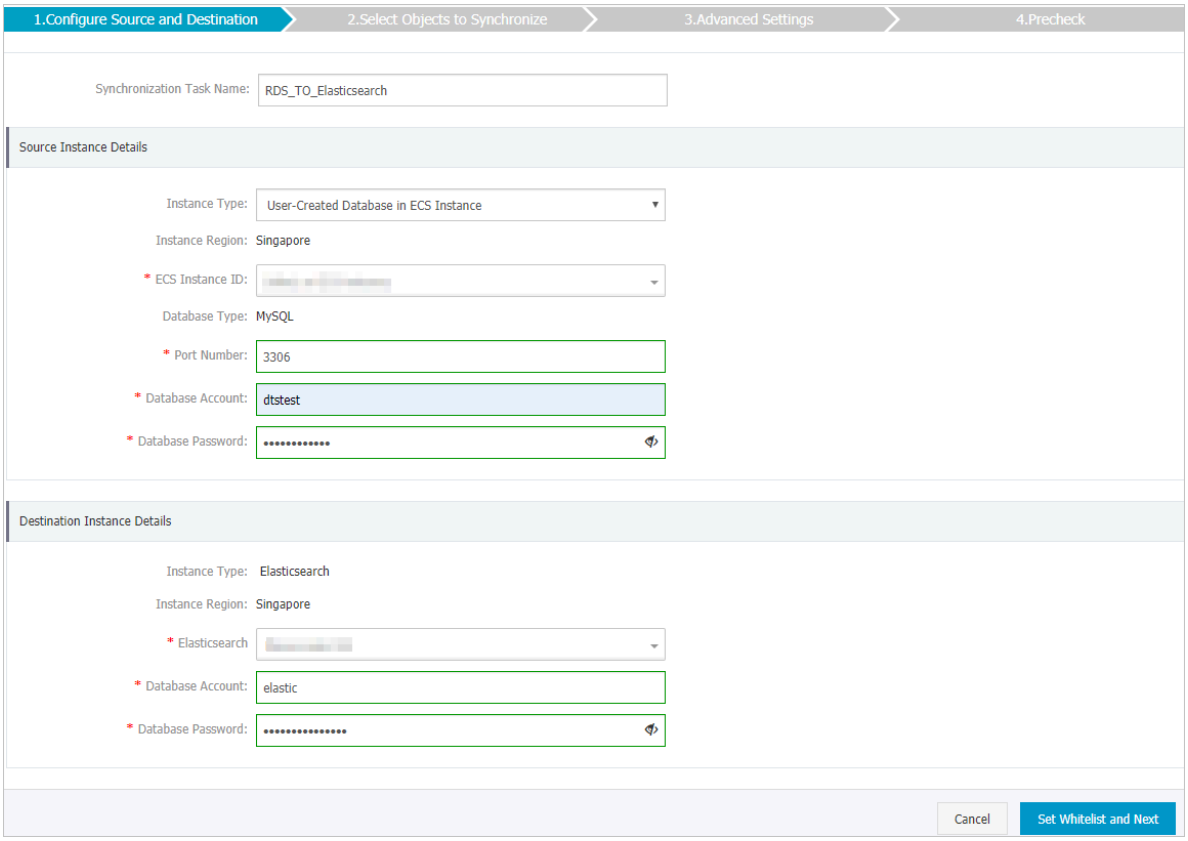
2. Log on to the DTS console.
3. In the left-side navigation pane, click **Data Synchronization**.

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	Synchronization 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.

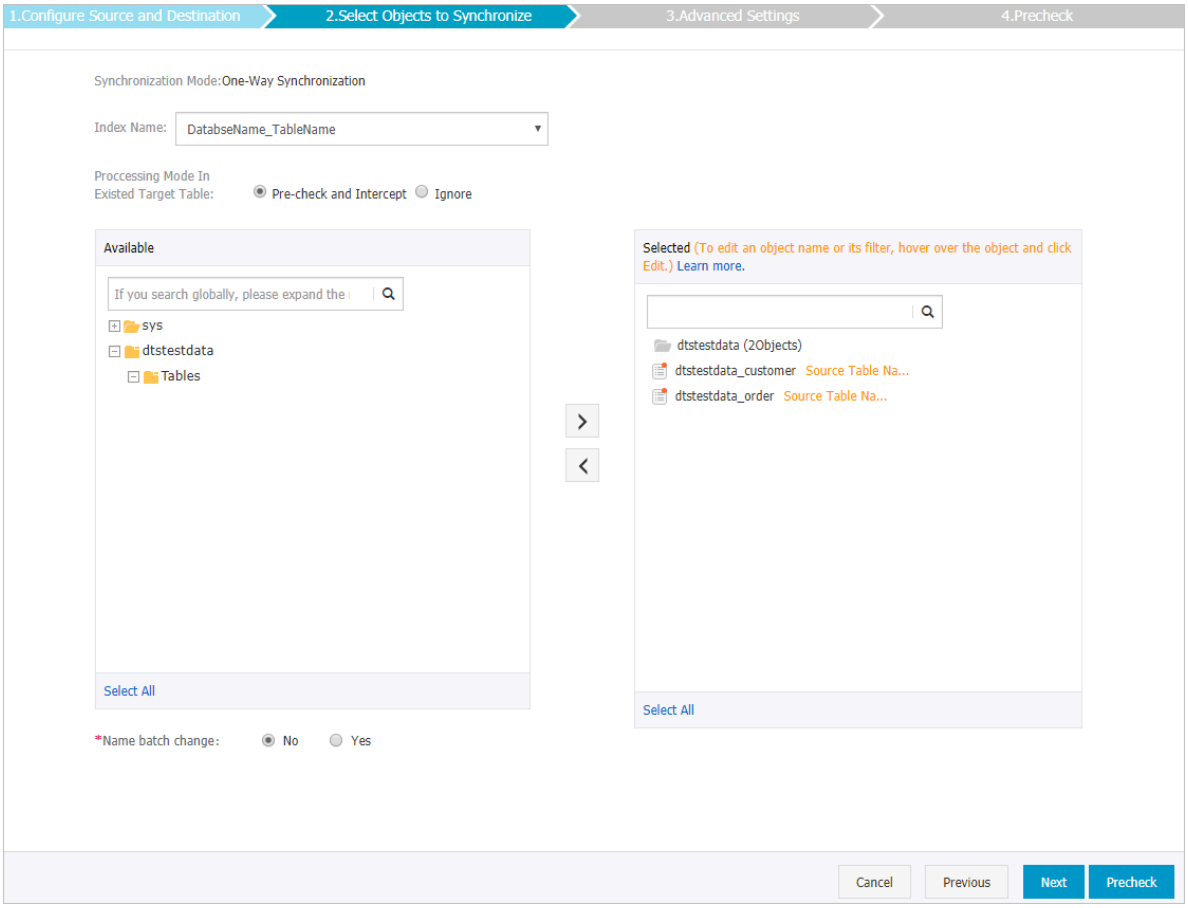
Section	Parameter	Description
Source Instance Details	Instance Type	Select **User-Created Database in ECS Instance**.
	Instance Region	The region of the source instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the 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.
	Database Account	Enter the account for 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.
	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.
	Elasticsearch	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.

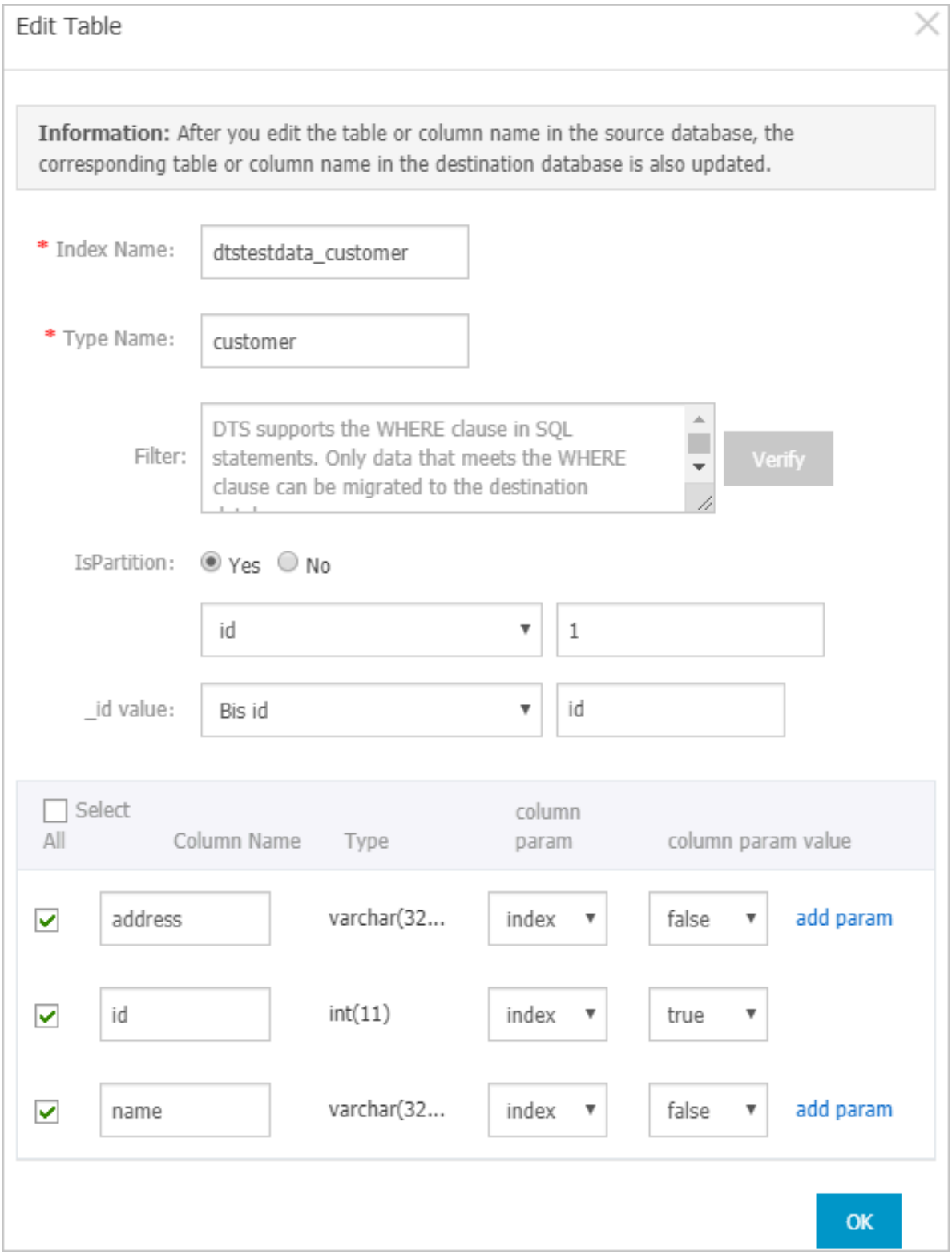
8. Configure the index name, the processing mode of identical index names, and the objects to be synchronized.



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.

Parameter	Description
Processing Mode In Existed Target Table	• **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.  **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**.
_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**.

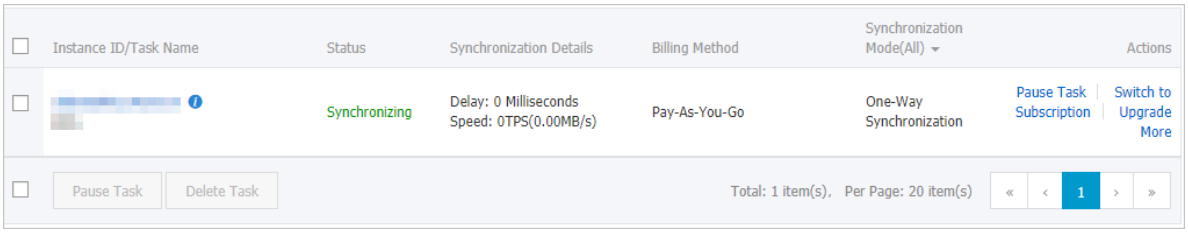
Note:

- 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.

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.

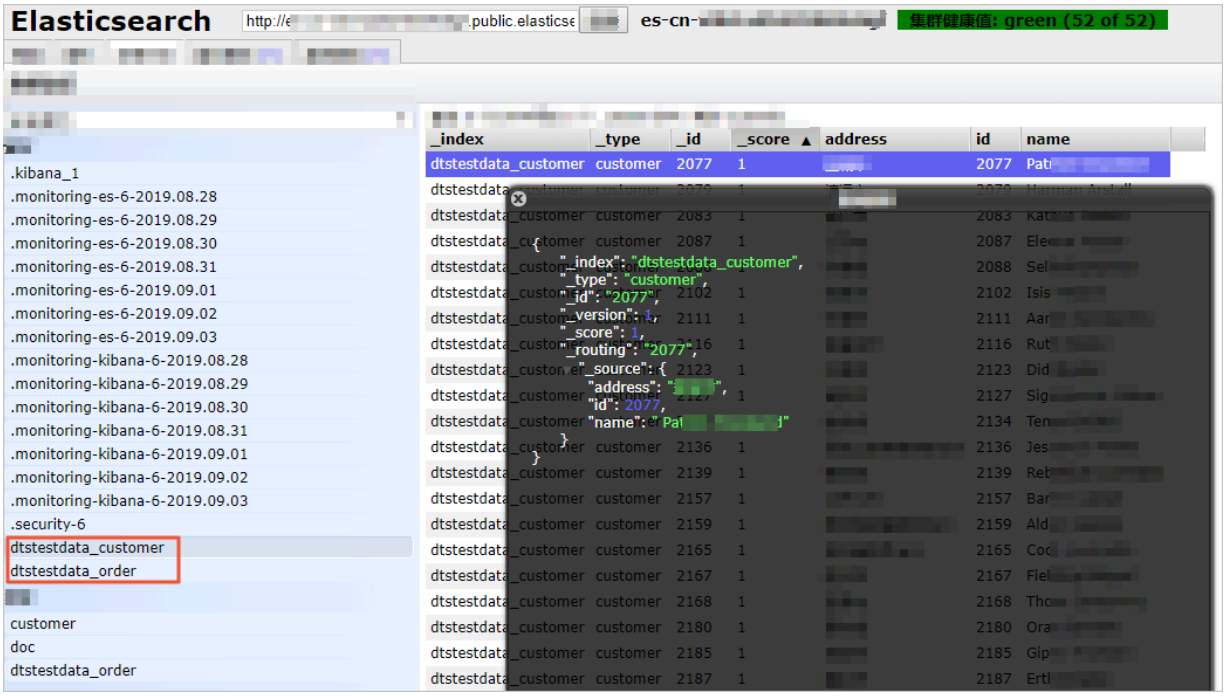
On the **Synchronization Tasks** page, view the status of the data synchronization task.



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 #unique_57.

SQL operations that can be synchronized

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

Synchronization process

1. 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.

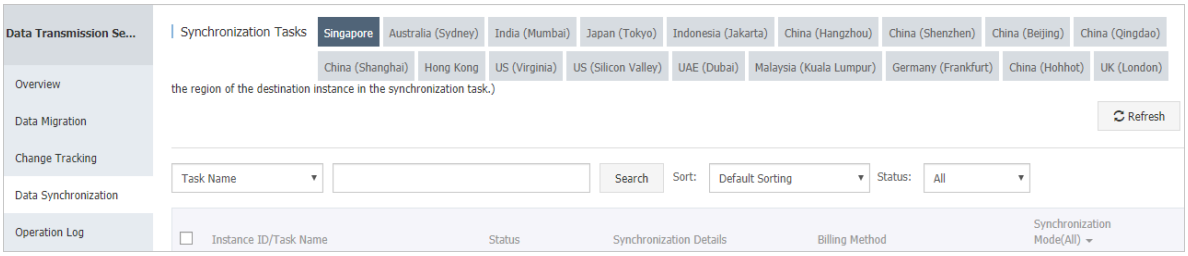
Procedure

> **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 #unique_51.

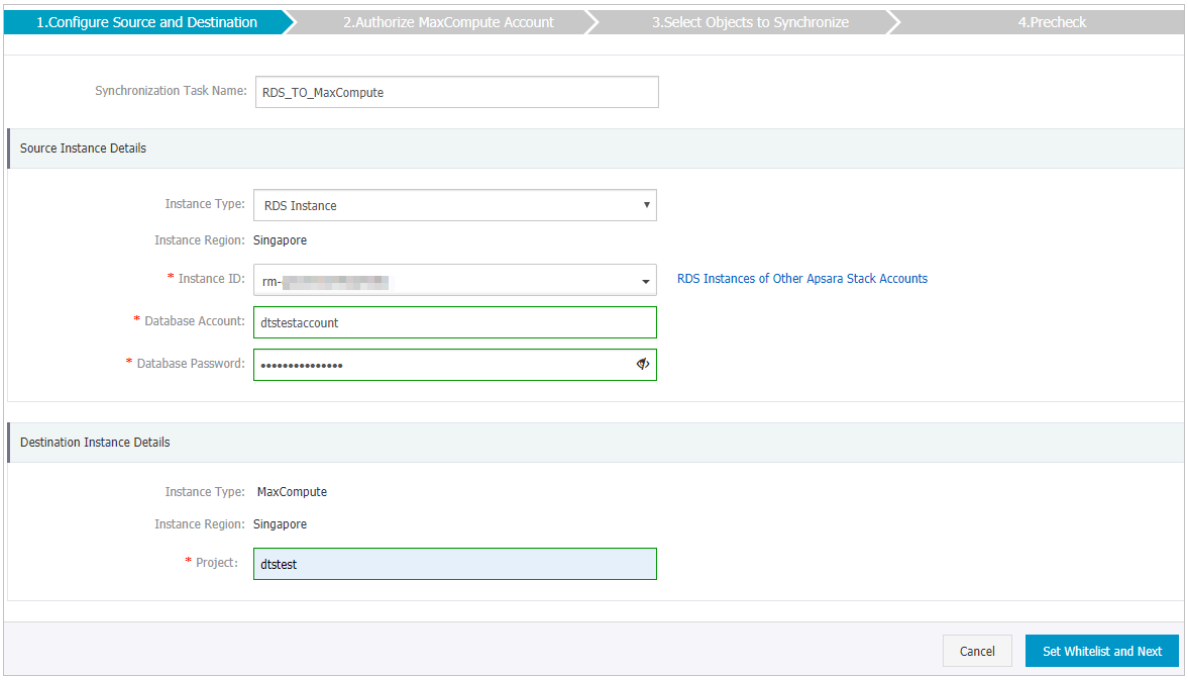
   > **Note:**
   > On the buy page, set Source Instance to **MySQL**, set Target Instance to **MaxCompute**, and set Synchronization Topology to **One-Way Synchronization**.

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.



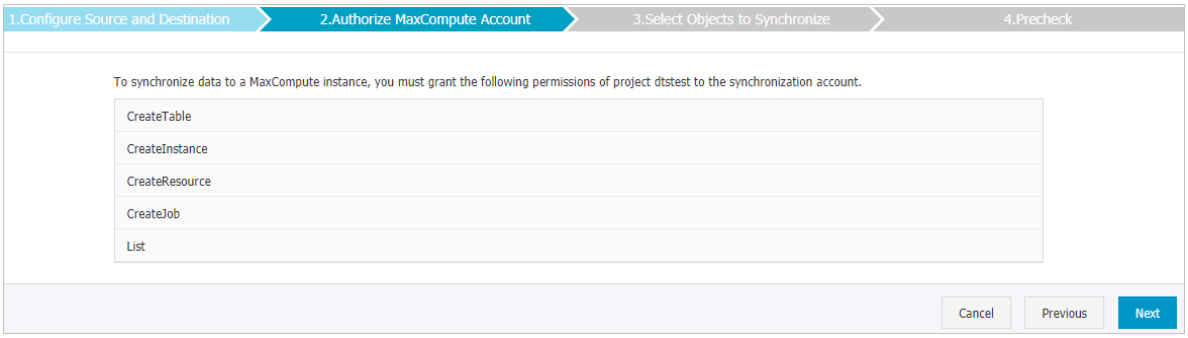
Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **RDS Instance**.
	Instance Region	The 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.
	Database Account	Enter the database account 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 source 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 you configure the data synchronization task. For more information, see Configure SSL encryption for an RDS MySQL instance. **Notice:** The **Encryption** parameter is available only for regions in mainland China and the Hong Kong (China) region.
Destination Instance Details	Instance Type	The value of this parameter is set to **MaxCompute** 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.
	Project	Enter the name of the MaxCompute **project**. You can search for a project on the Workspaces page in the DataWorks console.

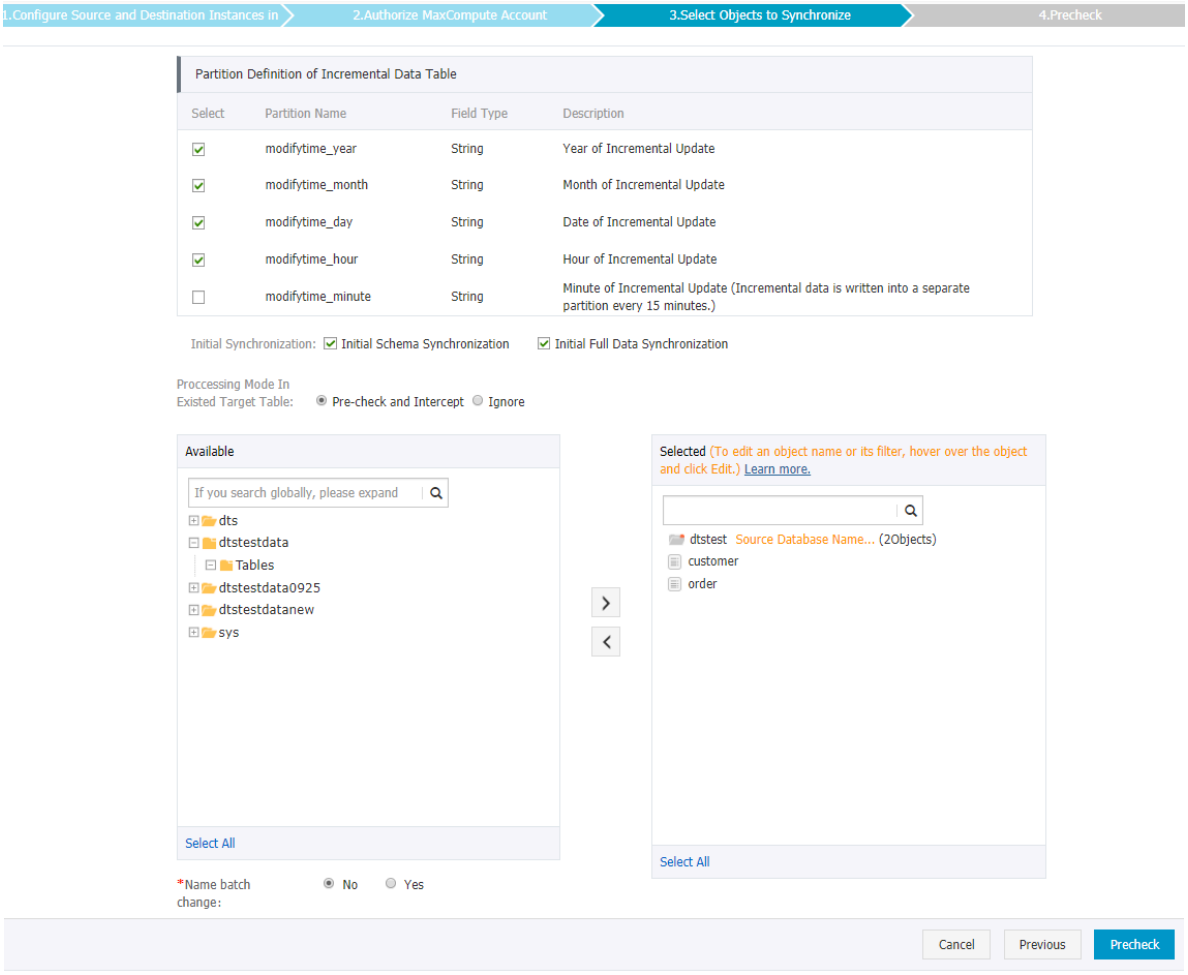
7. In the lower-right corner of the page, click **Set Whitelist and Next**.

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.



Parameter	Description
Partition Definition of Incremental Data Table	Select the partition name based on your business requirements. For more information about partitions, see Partition.
Initial Synchronization	Initial synchronization includes initial schema synchronization and initial full data synchronization. Select both **Initial Schema Synchronization** and **Initial Full Data Synchronization**. In this case, DTS synchronizes the schemas and historical data of the required objects from the source database to the destination database before synchronizing incremental data.

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. - 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.

Parameter	Description
Objects to be synchronized	Select tables from the **Available** section and click the icon to move the tables to the **Selected** section. **Note:** • You can select tables from multiple databases as the objects to be synchronized. • 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.

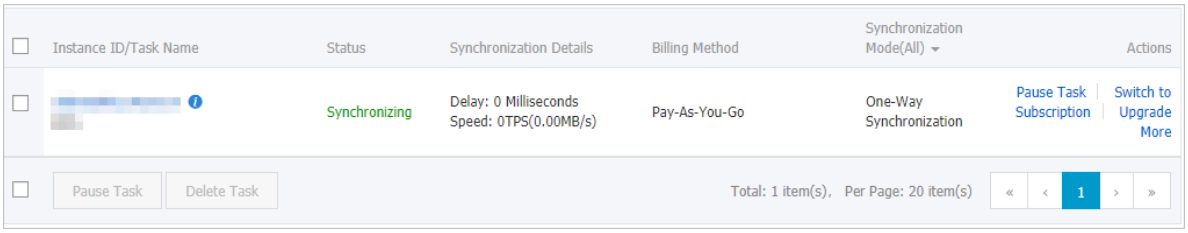
10. In the lower-right corner of the page, click **Precheck**.

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.

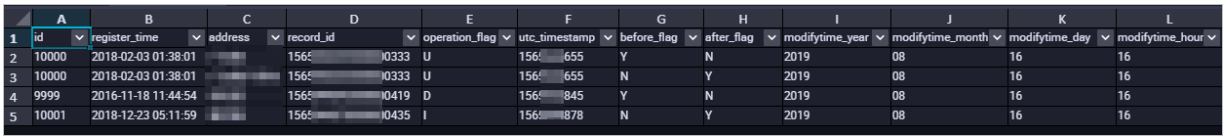
11. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.**
12. Wait until the initial synchronization is complete and the data synchronization task is in the **Synchronizing** state.

    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.



	A	B	C	D	E	F	G	H	I	J	K	L
1	id	register_time	address	record_id	operation_flag	utc_timestamp	before_flag	after_flag	modifytime_year	modifytime_month	modifytime_day	modifytime_hour
2	10000	2018-02-03 01:38:01		156[redacted]0333	U	156[redacted]655	Y	N	2019	08	16	16
3	10000	2018-02-03 01:38:01		156[redacted]0333	U	156[redacted]655	N	Y	2019	08	16	16
4	9999	2016-11-18 11:44:54		156[redacted]0419	D	156[redacted]845	Y	N	2019	08	16	16
5	10001	2018-12-23 05:11:59		156[redacted]0435	I	156[redacted]878	N	Y	2019	08	16	16

Note:

In the example, the modifytime_year, modifytime_month, modifytime_day, modifytime_hour, and modifytime_minute fields form the partition key. These fields are specified in the Configure the synchronization policy and objects step.

Schema of an incremental data table

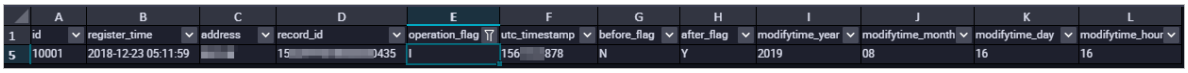
Field	Description
record_id	The ID of the incremental log entry. **Note:** • The ID auto-increments for each new log entry. • If an UPDATE operation is performed, DTS generates two incremental log entries for the operation. The two incremental log entries have the same record ID.
operation_flag	The operation 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.

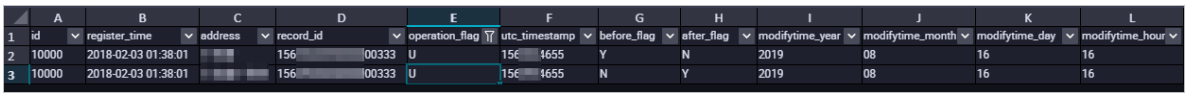


	A	B	C	D	E	F	G	H	I	J	K	L
1	id	register_time	address	record_id	operation_flag	utc_timestamp	before_flag	after_flag	modifytime_year	modifytime_month	modifytime_day	modifytime_hour
5	10001	2018-12-23 05:11:59		15 0435	I	156 878	N	Y	2019	08	16	16

- 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.

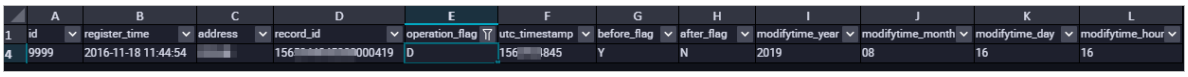
  The second log entry records the pre-update values, so the value of the before_flag field is Y and the value of the after_flag field is N. The second log entry records the post-update values, so the value of the before_flag field is N and the value of the after_flag field is Y.



	A	B	C	D	E	F	G	H	I	J	K	L
1	id	register_time	address	record_id	operation_flag	utc_timestamp	before_flag	after_flag	modifytime_year	modifytime_month	modifytime_day	modifytime_hour
2	10000	2018-02-03 01:38:01		156 00333	U	15 4655	Y	N	2019	08	16	16
3	10000	2018-02-03 01:38:01		156 00333	U	15 4655	N	Y	2019	08	16	16

- 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.



	A	B	C	D	E	F	G	H	I	J	K	L
1	id	register_time	address	record_id	operation_flag	utc_timestamp	before_flag	after_flag	modifytime_year	modifytime_month	modifytime_day	modifytime_hour
4	9999	2016-11-18 11:44:54		156 000419	D	156 845	Y	N	2019	08	16	16

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.

	Field	Type	Null	Key	Default	Extra
1	id	int(11)	NO	PRI	null	
2	register_time	timestamp	YES		null	
3	address	varchar(32)	YES		null	

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, <col1>, <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>
 union all
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'
```

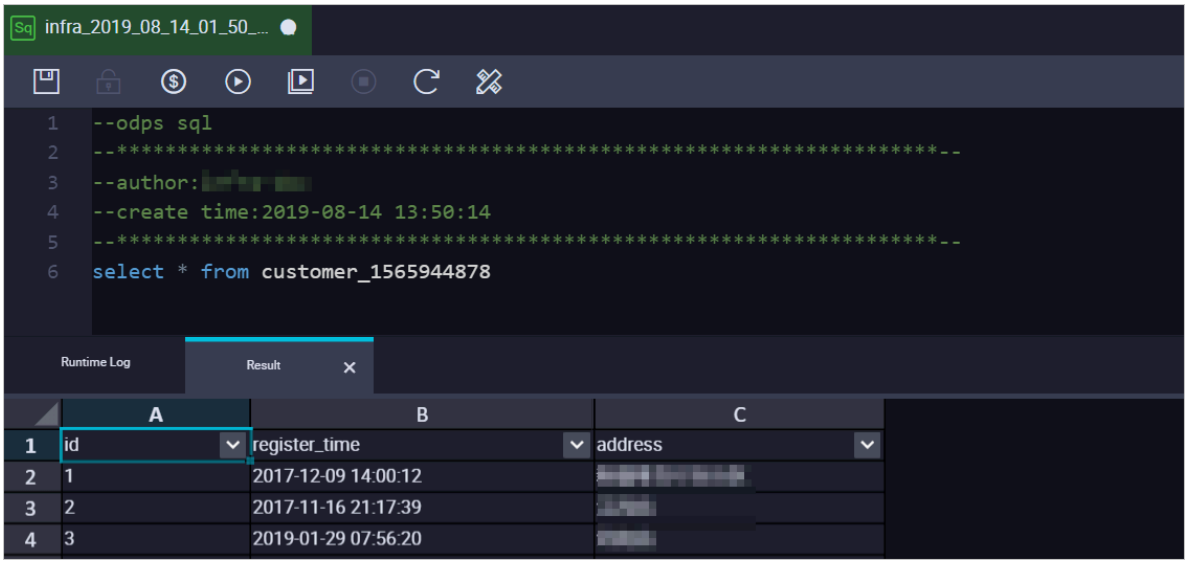
   Note:

- <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 1565944878 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
  from(
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
 union all
select 0 as record_id, 'I' as operation_flag, 'Y' as after_flag, base.id, base.register_time
, base.address
  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.



8 Synchronize data to or from an Apsara PolarDB cluster

8.1 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 gh-ost 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

Operation 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

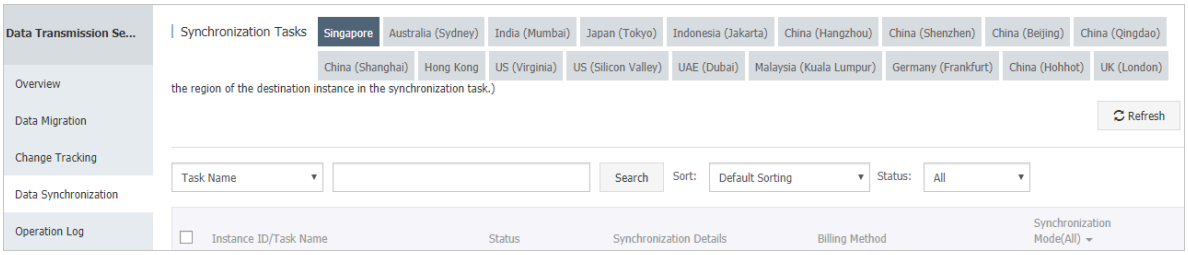
  RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A needs to be synchronized and it is renamed Table B, Table B cannot be synchronized to the destination database. To avoid this situation, you can select the database to which 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 #unique_51.

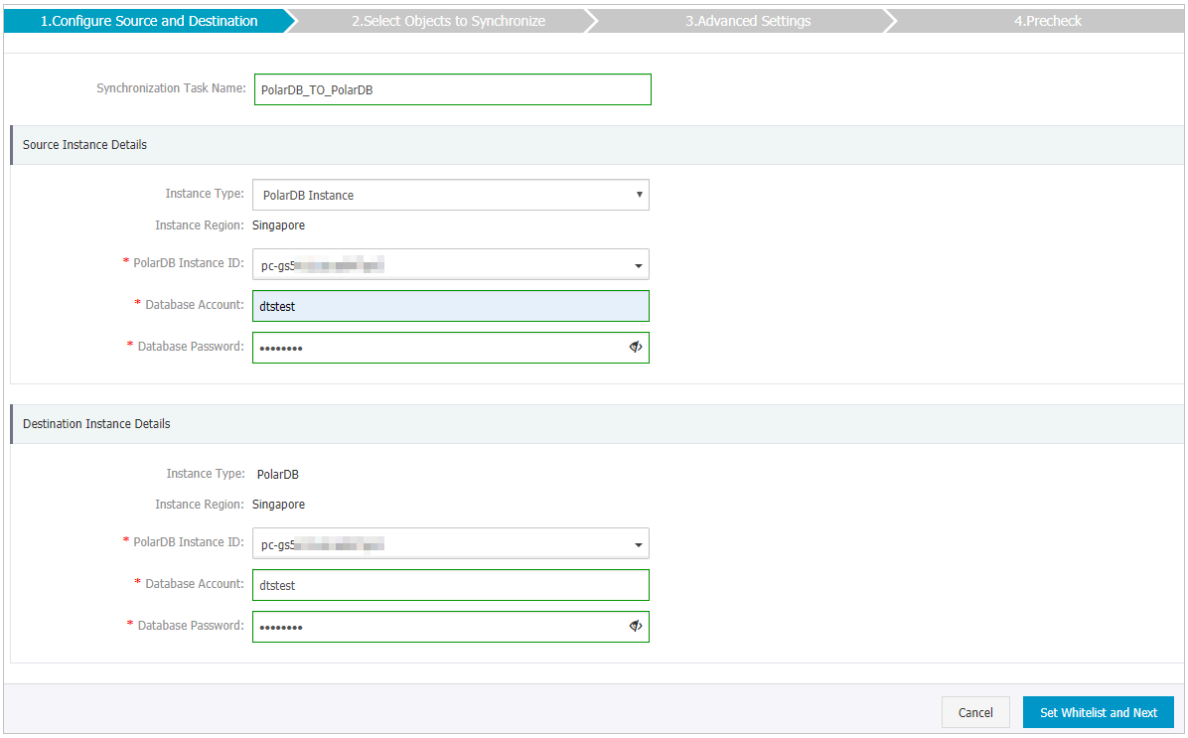
   Note:
   On the buy page, set both Source Instance and Target Instance to **POLARDB**, and set Synchronization Topology to **One-Way Synchronization**.

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.



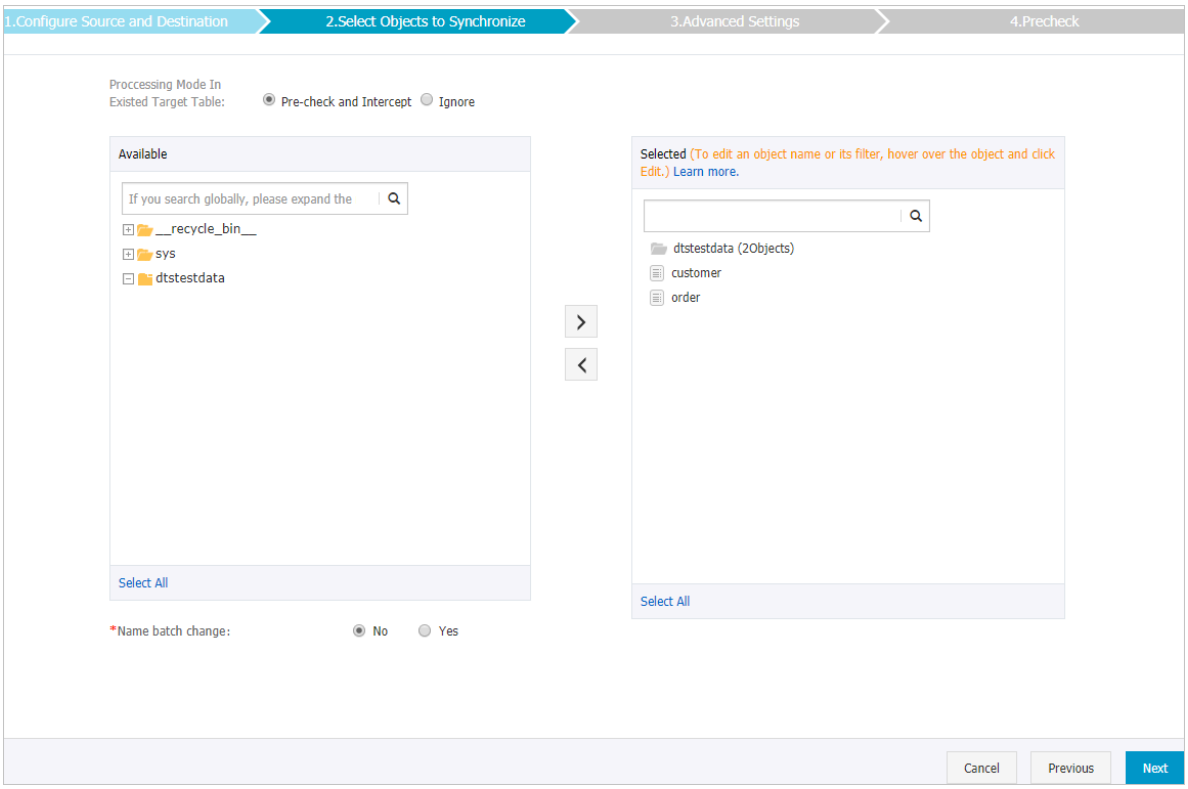
Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	The value of this parameter is set to **PolarDB 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.
	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.

Section	Parameter	Description
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 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.

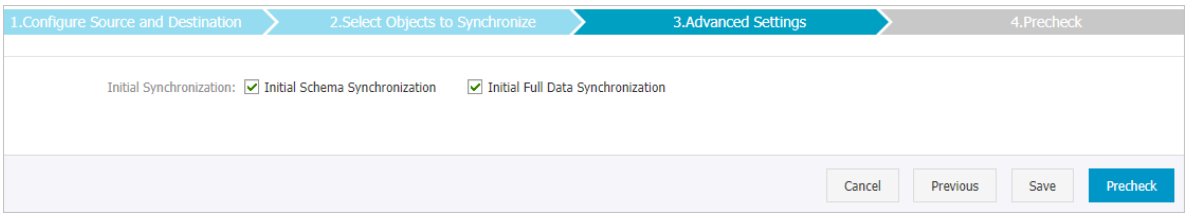
8. Configure the processing mode in existing destination tables and the objects to be synchronized.



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. - 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.

Parameter	Description
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.



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.

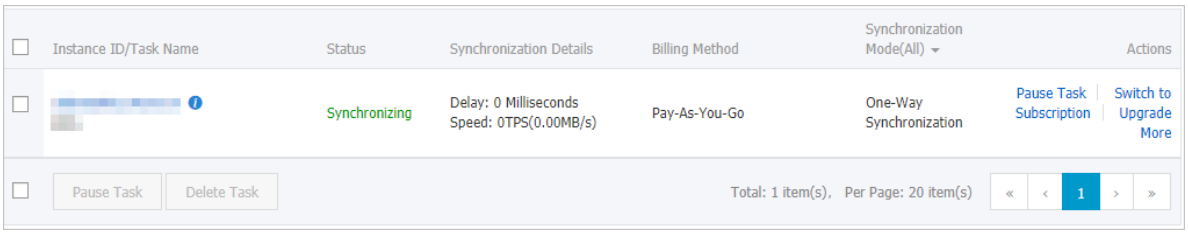
11. In the lower-right corner of the page, click **Precheck**.

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.



8.2 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

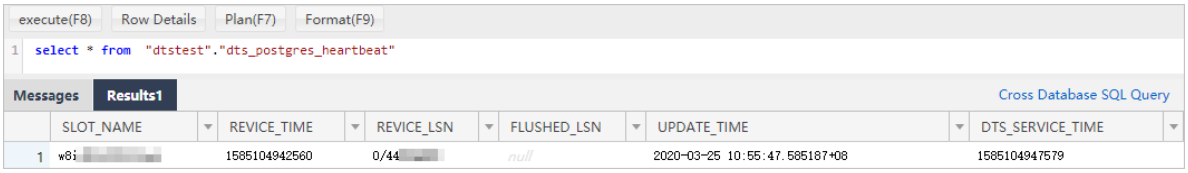
The tables to be synchronized contain primary keys or UNIQUE NOT NULL indexes.

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 #unique_59.

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

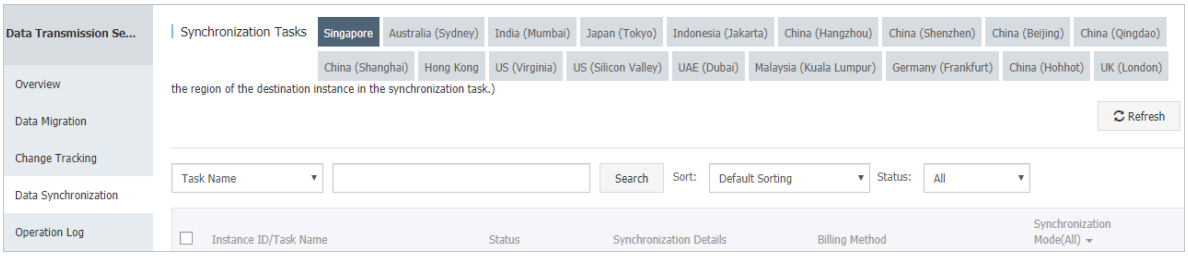
Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

   Note:
   On the buy page, set both Source Instance and Target Instance to **PolarDB**, and set Synchronization Topology to **One-Way Synchronization**.

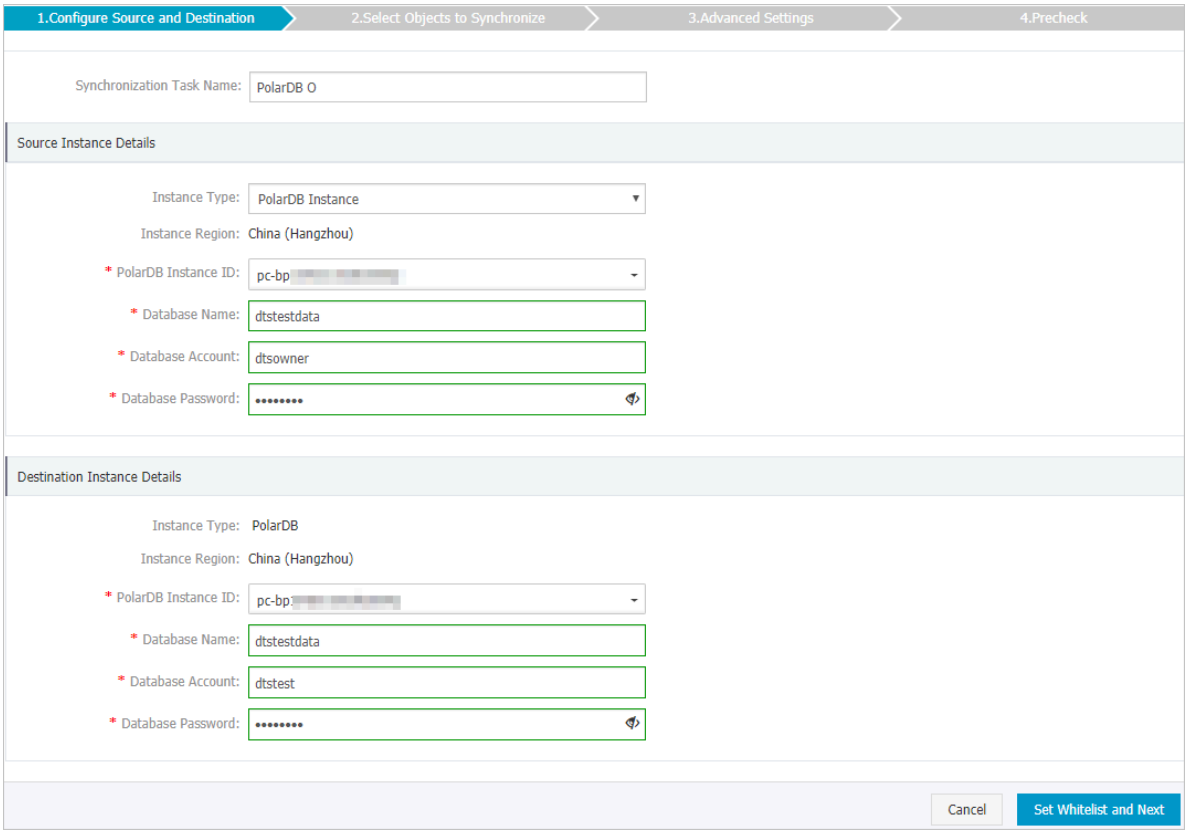
2. Log on to the DTS console.
3. In the left-side navigation pane, click **Data Synchronization**.

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	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.

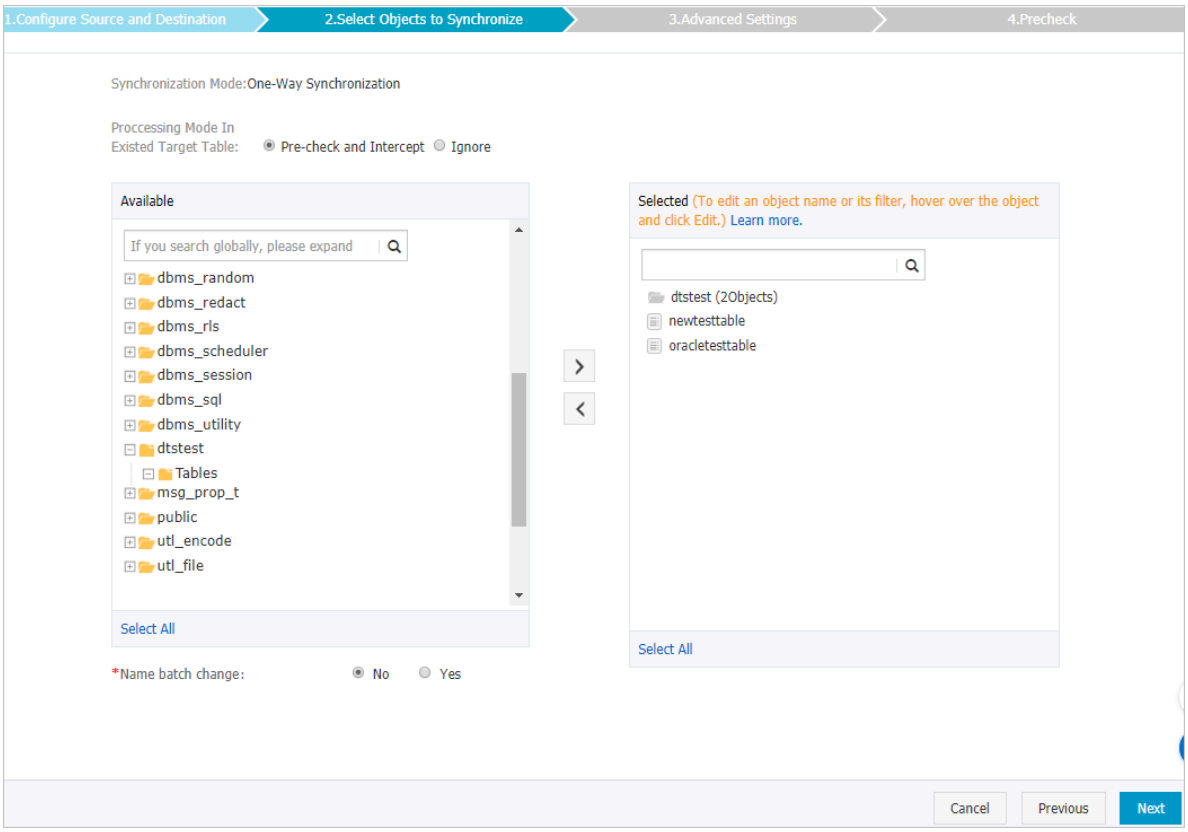
Section	Parameter	Description
Source Instance Details	Instance Type	The value of this parameter is set to **PolarDB Instance** and cannot be changed.
	Instance Region	The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	PolarDB Instance ID	Select the ID of the source PolarDB cluster.
	Database 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 #unique_60.
	Database Password	Enter the password of the source database account.
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.

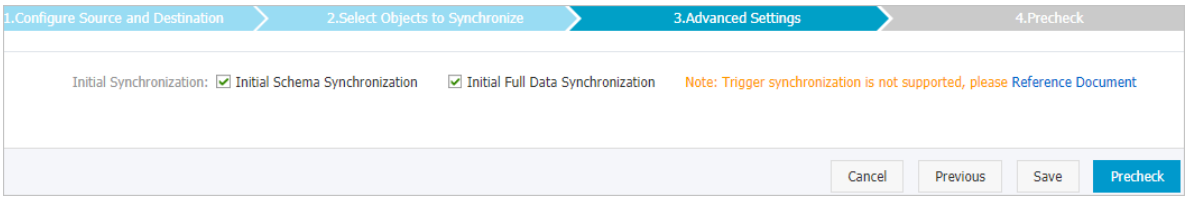
8. Configure the processing mode in existing destination tables and the objects to be synchronized.



Parameter	Description
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. - 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.

Parameter	Description
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.



Initial synchronization type	Description
Initial schema synchronization	DTS synchronizes the schemas of the required objects to the destination PolarDB cluster. DTS supports initial schema synchronization for the following types of objects: table, view, synonym, trigger, stored procedure, function, package, and user-defined type. **Notice:** However, if an object contains triggers, data will become inconsistent between the source and destination databases. For more information about how to solve this issue, see #unique_50.

Initial synchronization type	Description
Initial full data synchronization	DTS synchronizes historical data of the required objects from the source PolarDB cluster to the destination PolarDB cluster. **Notice:** During initial schema synchronization and initial full data synchronization, do not perform DDL operations on the objects to be synchronized. Otherwise, the objects may fail to be synchronized.

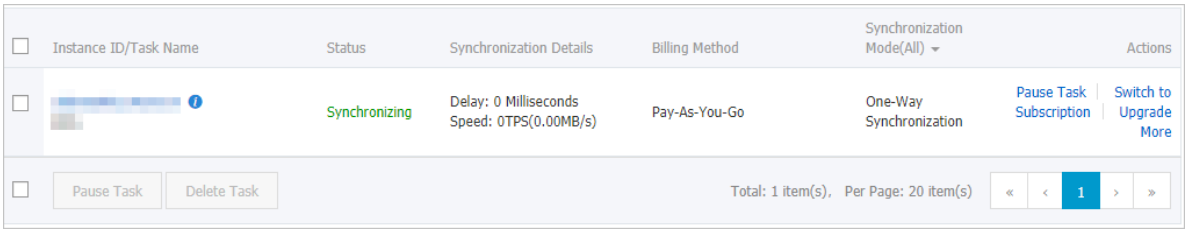
11. In the lower-right corner of the page, click **Precheck**.

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.



8.3 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 gh-ost 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

Operation 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

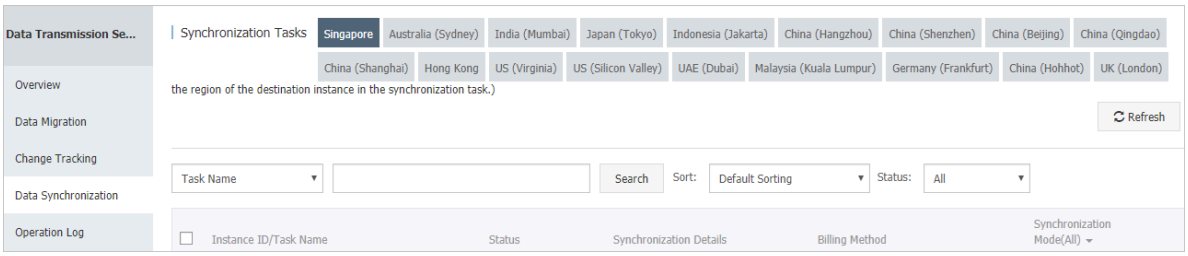
this situation, you can select the database to which Table A and Table B belong as the object when configuring the data synchronization task.

Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

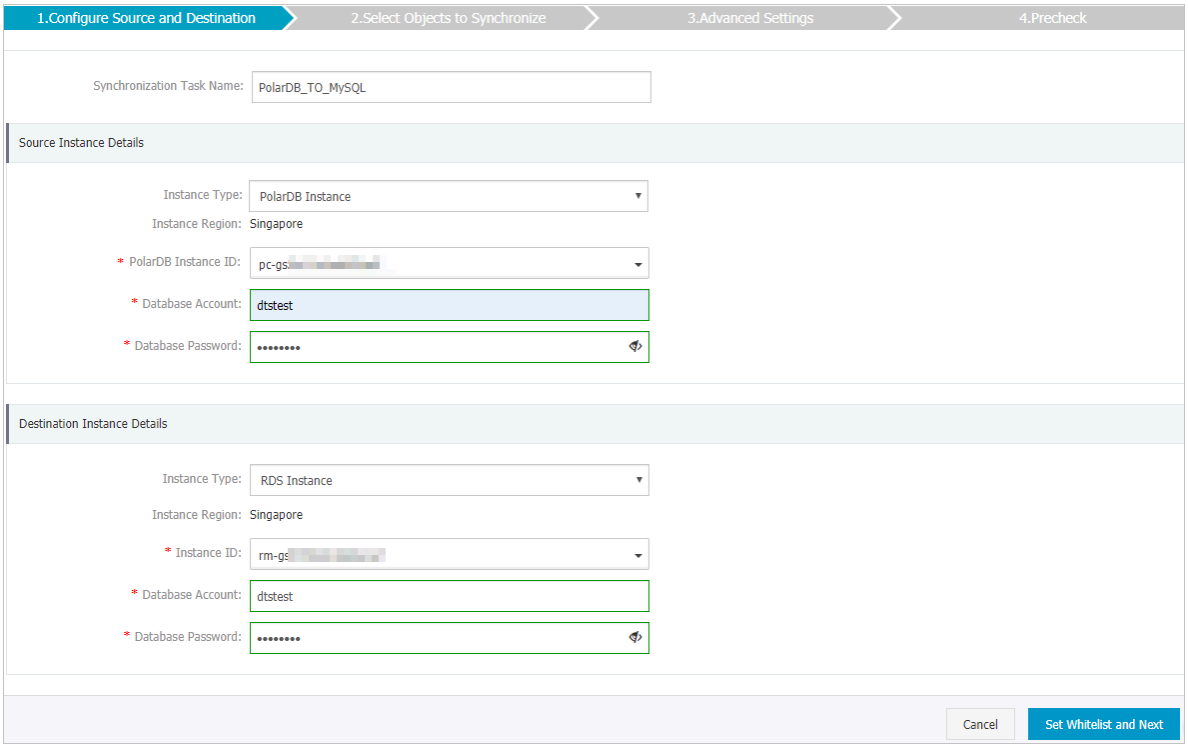
   > **Note:**
   > On the buy page, set Source Instance to **PolarDB**, Target Instance to **MySQL**, and Synchronization Topology to **One-Way Synchronization**.

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.



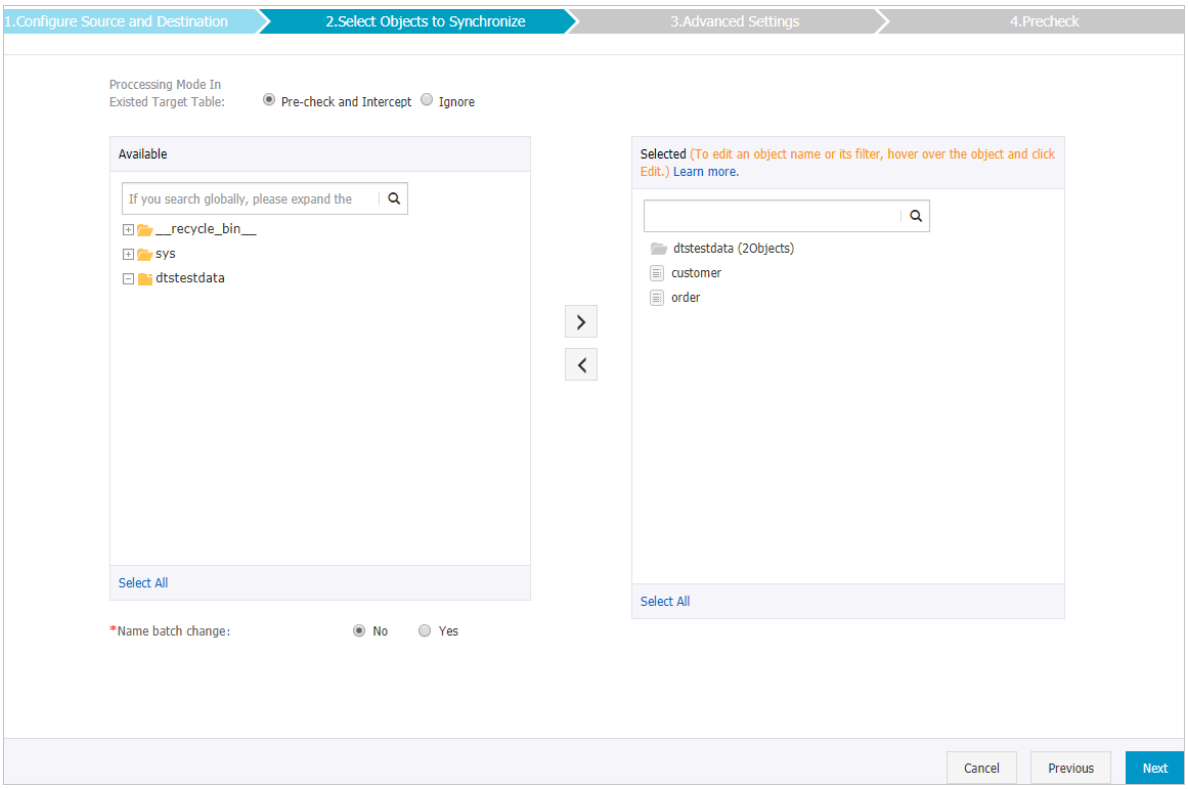
Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	The value of this parameter is set to **PolarDB 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.
	Database Account	Enter the database account of the source PolarDB cluster.
	Database Password	Enter the password for the source database account.

Section	Parameter	Description
Destination Instance Details	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. **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).

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 PolarDB cluster and the destination RDS instance. This ensures that DTS servers can connect to the source PolarDB cluster and the destination RDS instance.

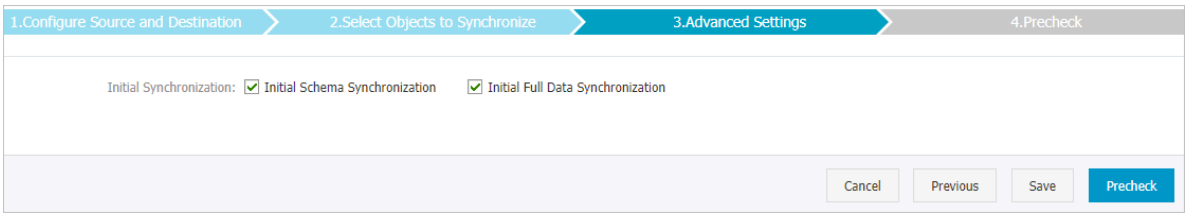
8. Configure the processing mode in existing destination tables and the objects to be synchronized.



Parameter	Description
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. - 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.

Parameter	Description
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.



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.

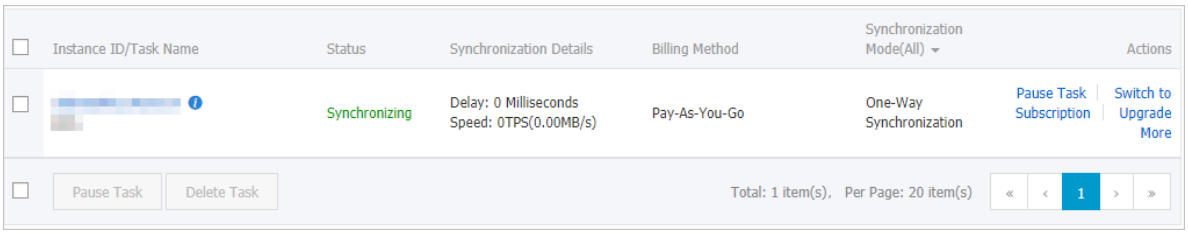
11. In the lower-right corner of the page, click **Precheck**.

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.



8.4 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

Database	Required permission
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

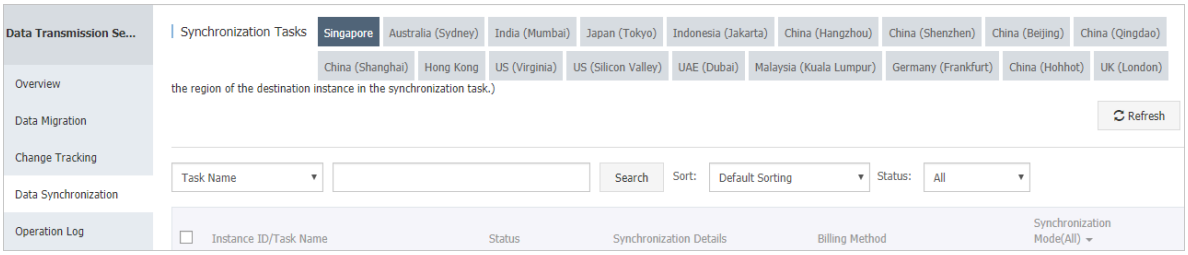
For more information, see Data type mappings for initial schema synchronization.

Procedure

1. Purchase a data synchronization instance. For more information, see#unique_51.

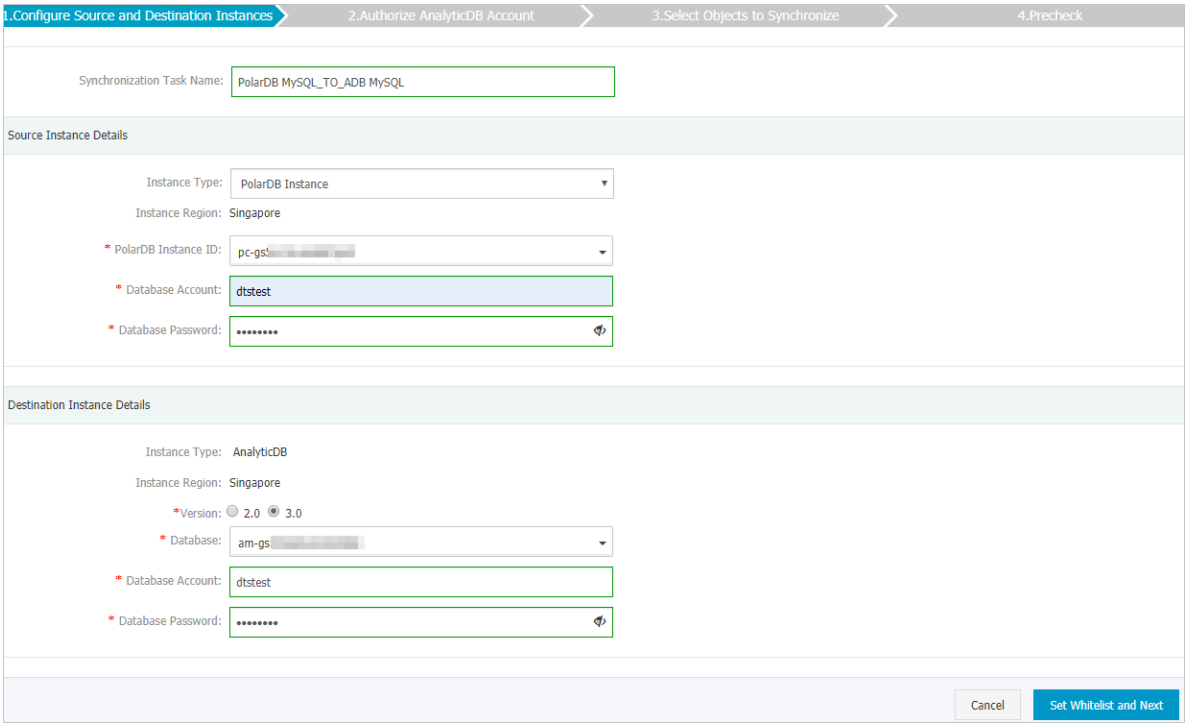
   Note:
   On the buy page, set Source Instance to **PolarDB**, Target Instance to **AnalyticDB for MySQL**, and Synchronization Topology to **One-Way Synchronization**.

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	Synchronization 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.
	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.

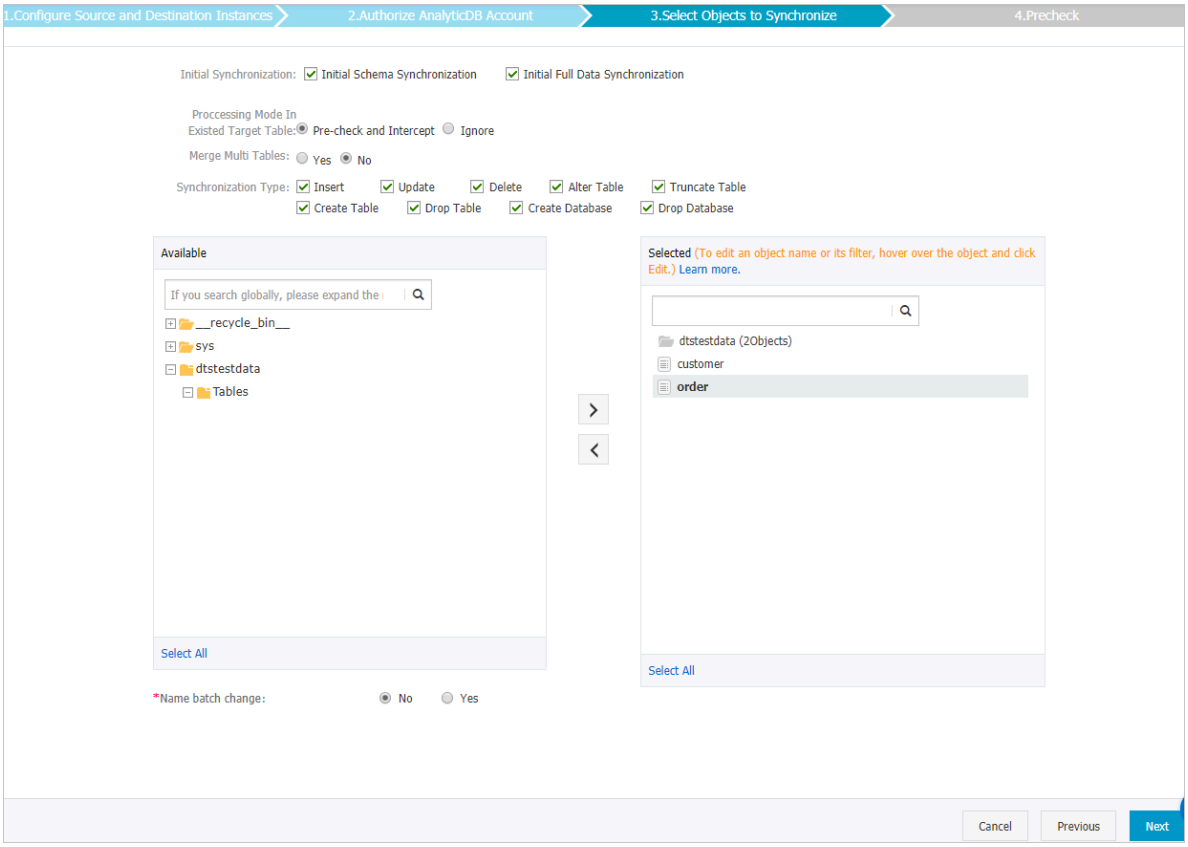
Section	Parameter	Description
Destination Instance Details	Instance Type	The value of this parameter is set to **AnalyticDB** and cannot be changed.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data 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.

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 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.

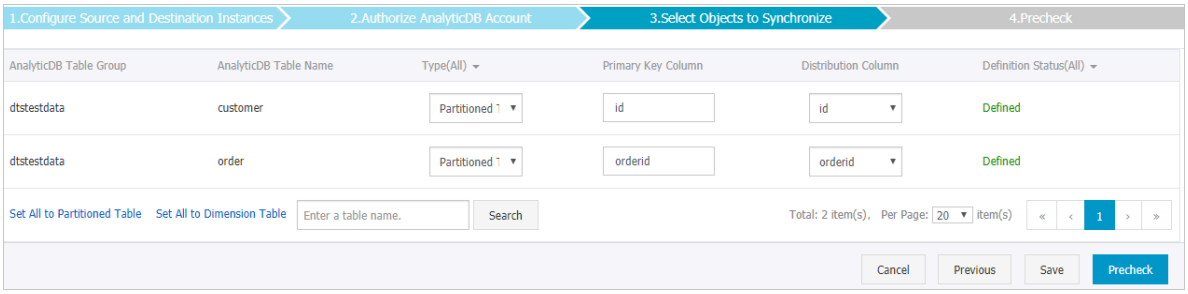


Parameter	Description
Initial Synchronization	You must select both **Initial Schema Synchronization** and **Initial Full Data Synchronization** in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination cluster. The schemas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

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, 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 the `__dts_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.

Parameter	Description
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.
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. • If you select a table as the object to be synchronized, only ADD COLUMN operations on the 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 name of an object in the destination 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. 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.

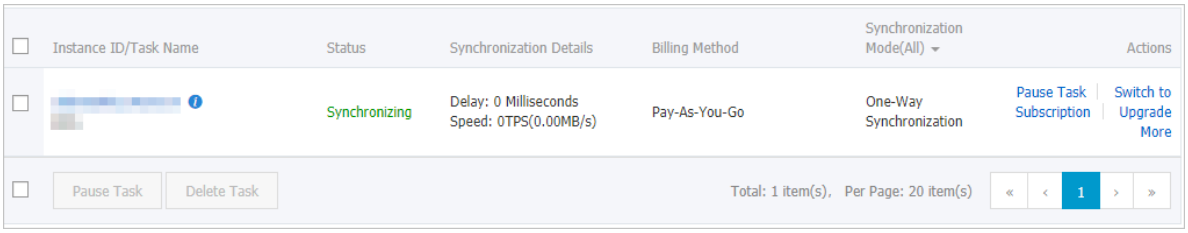
11. In the lower-right corner of the page, click **Precheck**.

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.

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 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

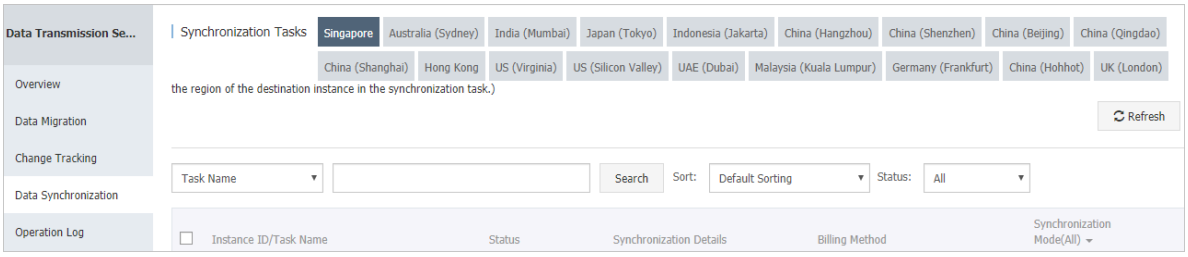
Term in Apsara PolarDB for MySQL	Term in AnalyticDB for PostgreSQL
Database	Schema
Table	Table

Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

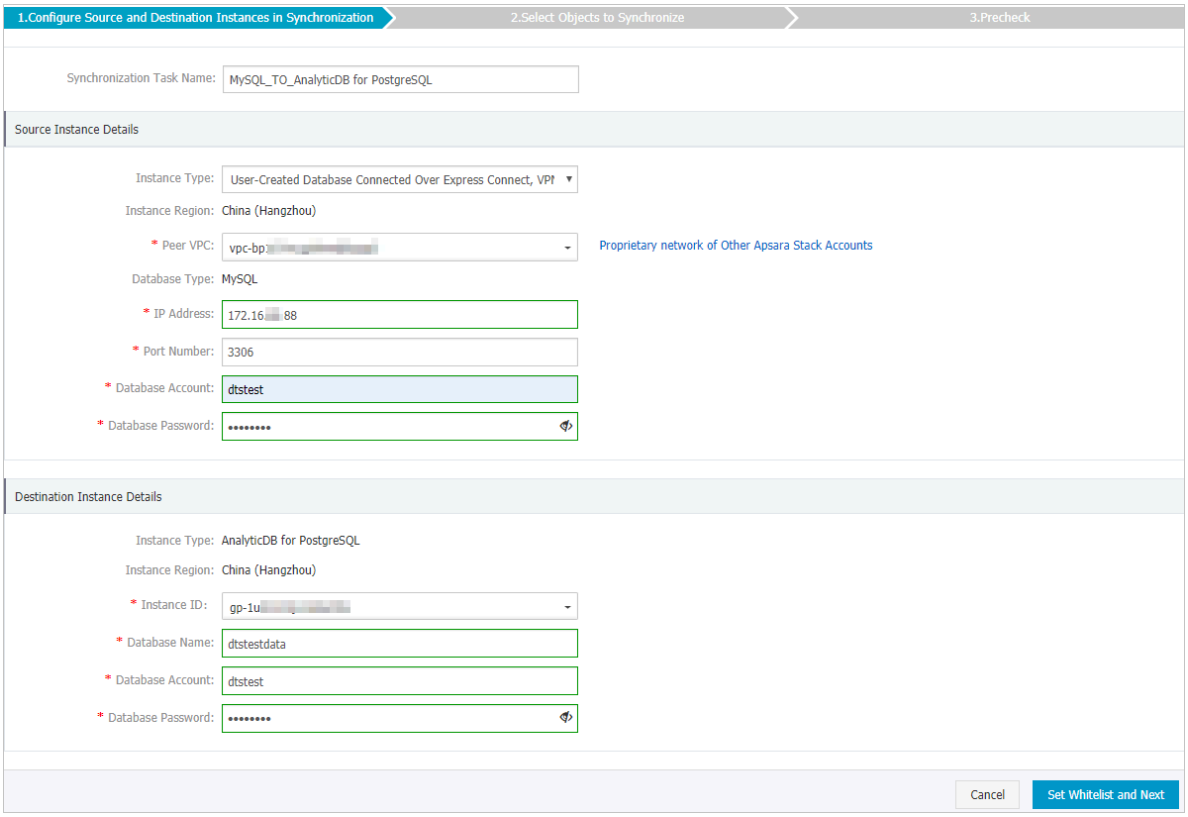
   Note:
   On the purchase page, select **MySQL** for the source instance and **AnalyticDB for PostgreSQL** for the destination instance. Select **One-Way Synchronization** as the synchronization topology.

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.

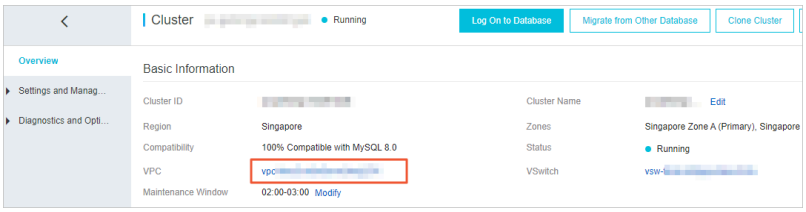
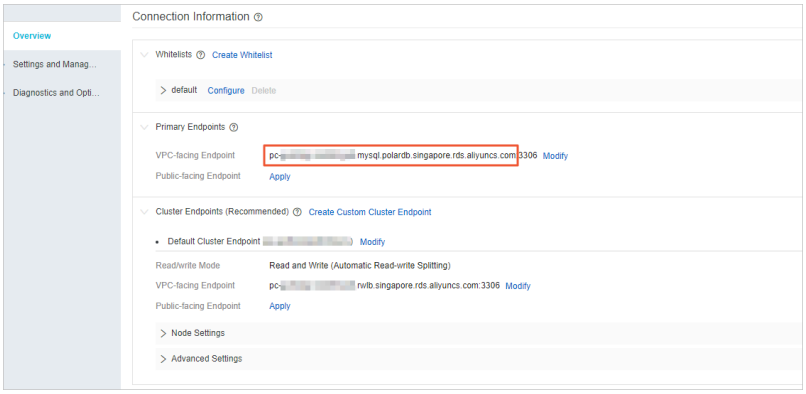


5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.

6. Configure the source and destination instances.



Section	Parameter	Description
N/A	Synchronization 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 Connected Over Express Connect, VPN Gateway, or Smart Access Gateway**. **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.
	Instance Region	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.

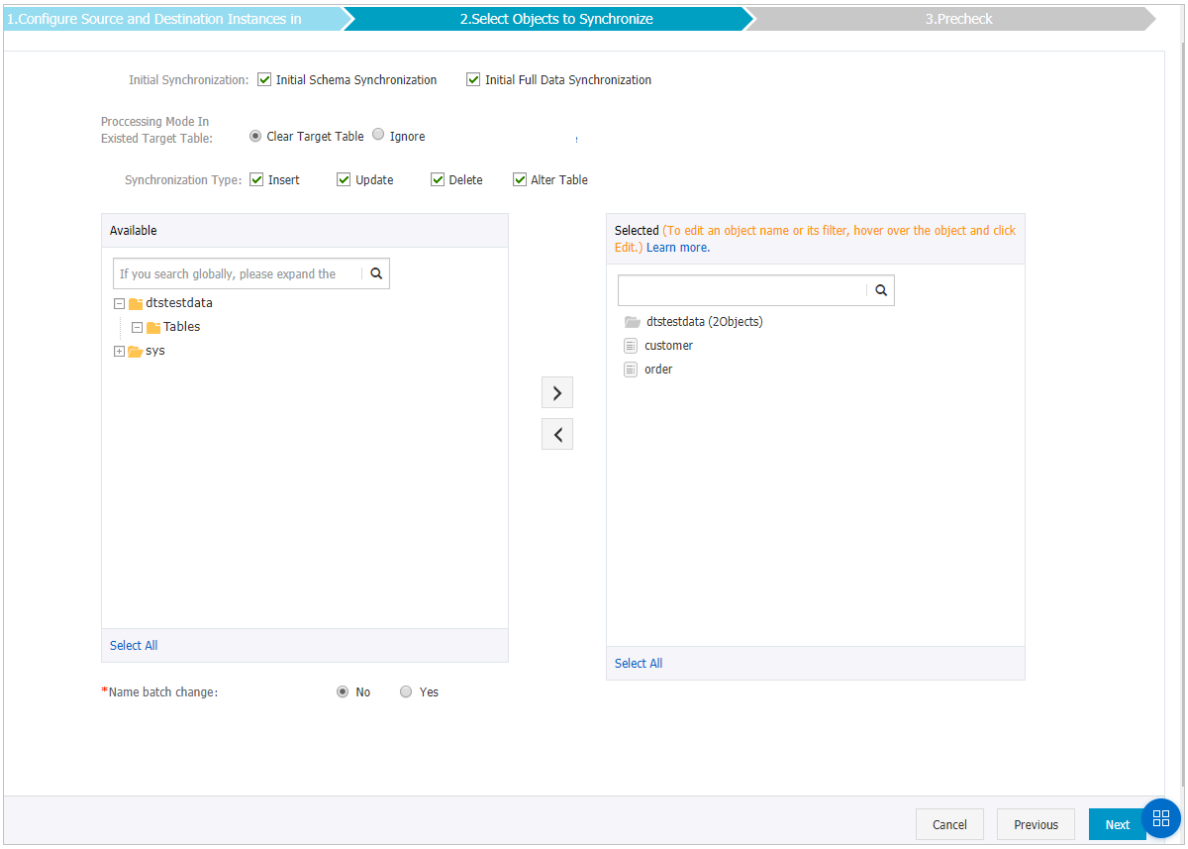
Section	Parameter	Description
	Peer VPC	Select the ID of the VPC where the Apsara PolarDB for MySQL cluster resides. To obtain the VPC ID, you can log on to the Apsara PolarDB console and click the cluster ID. On the Overview page that appears, you can view the ID of the VPC where the cluster resides in the **Basic Information** section. 
	Database Type	The value of this parameter is set to **MySQL** and cannot be changed.
	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. 
	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.

Section	Parameter	Description
	Database Password	Enter the password for the database account.
Destination Instance Details	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.
	Database Name	Enter the name of the destination database in the AnalyticDB for PostgreSQL instance.
	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.

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 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.

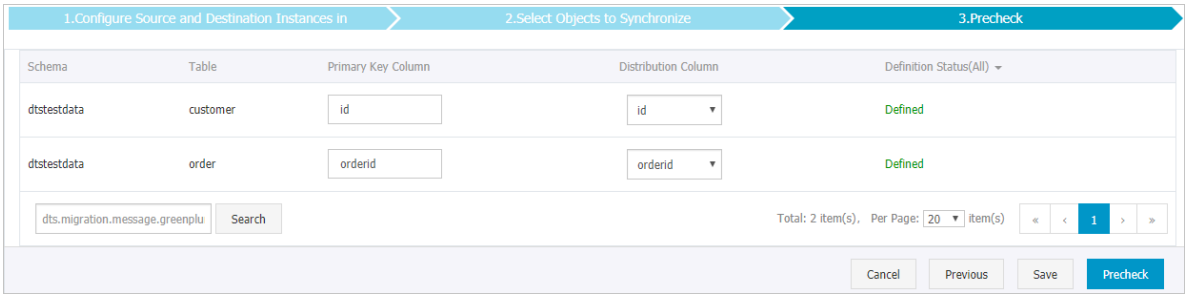


Section	Parameter	Description
Synchronization policy	Initial Synchronization	You must select both **Initial Schema Synchronization** and **Initial Full Data Synchronization** in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the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

Section	Parameter	Description
	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.
	Synchronization Type	Select the types of operations that you want to synchronize based on your business requirements. • **Insert** • **Update** • **Delete** • **Alter Table**

Section	Parameter	Description
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.



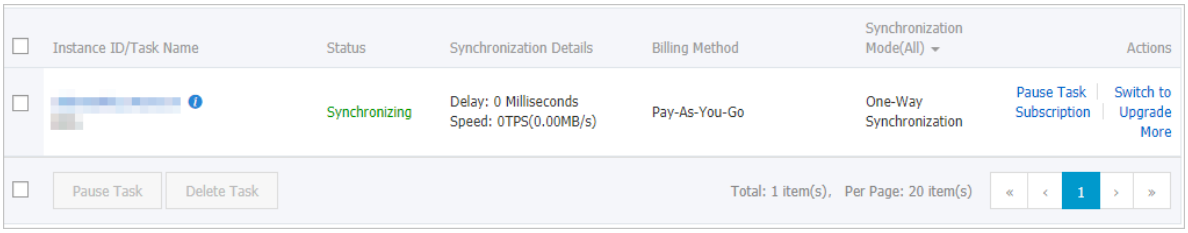
Note:
The page in this step appears only if you select **Initial Schema Synchronization**.

10. 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.

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.



8.6 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 gh-ost 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

Operation 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

#unique_52

> **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.

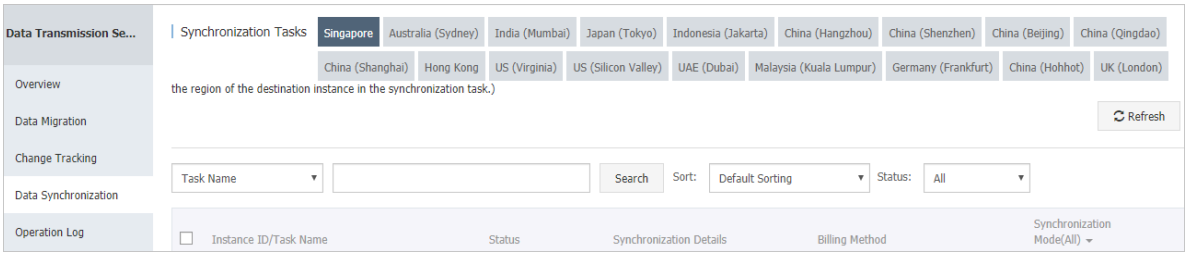
Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

   > **Note:**
   > On the buy page, set Source Instance to **MySQL**, Target Instance to **PolarDB**, and Synchronization Topology to **One-Way Synchronization**.

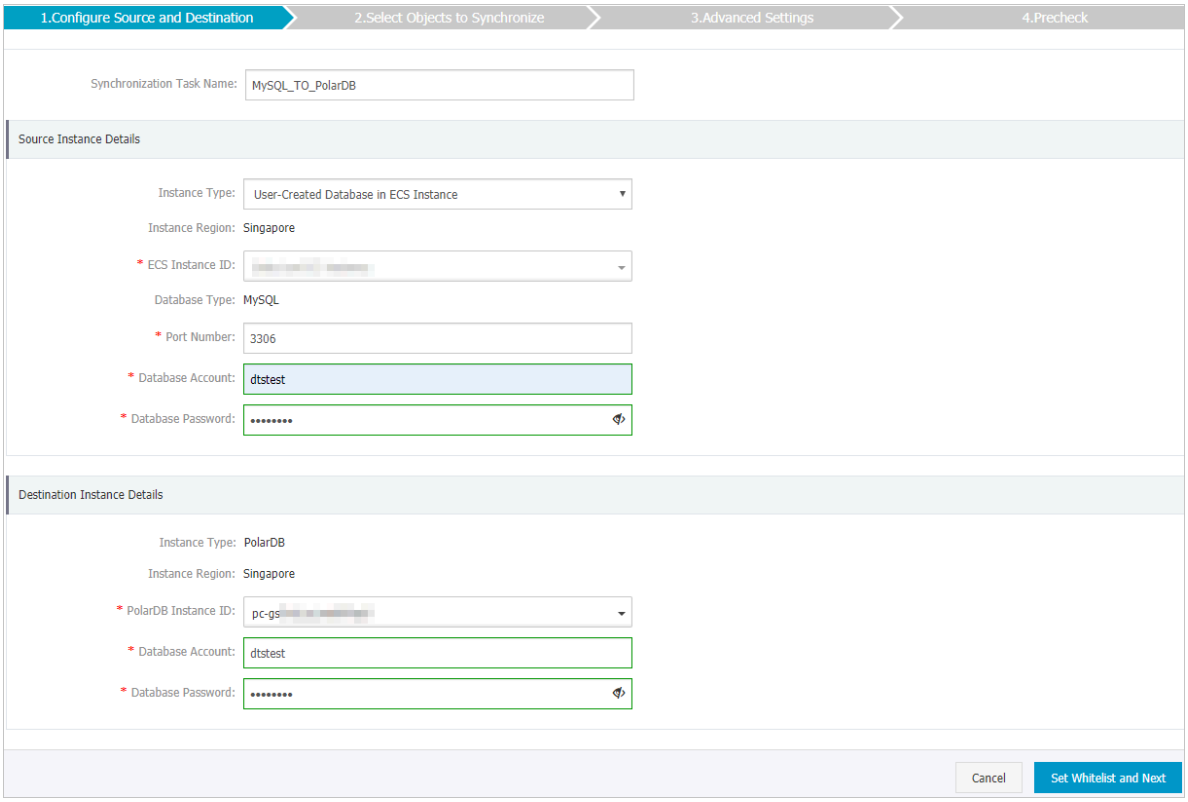
2. Log on to the DTS console.
3. In the left-side navigation pane, click **Data Synchronization**.

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	Synchronization 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.

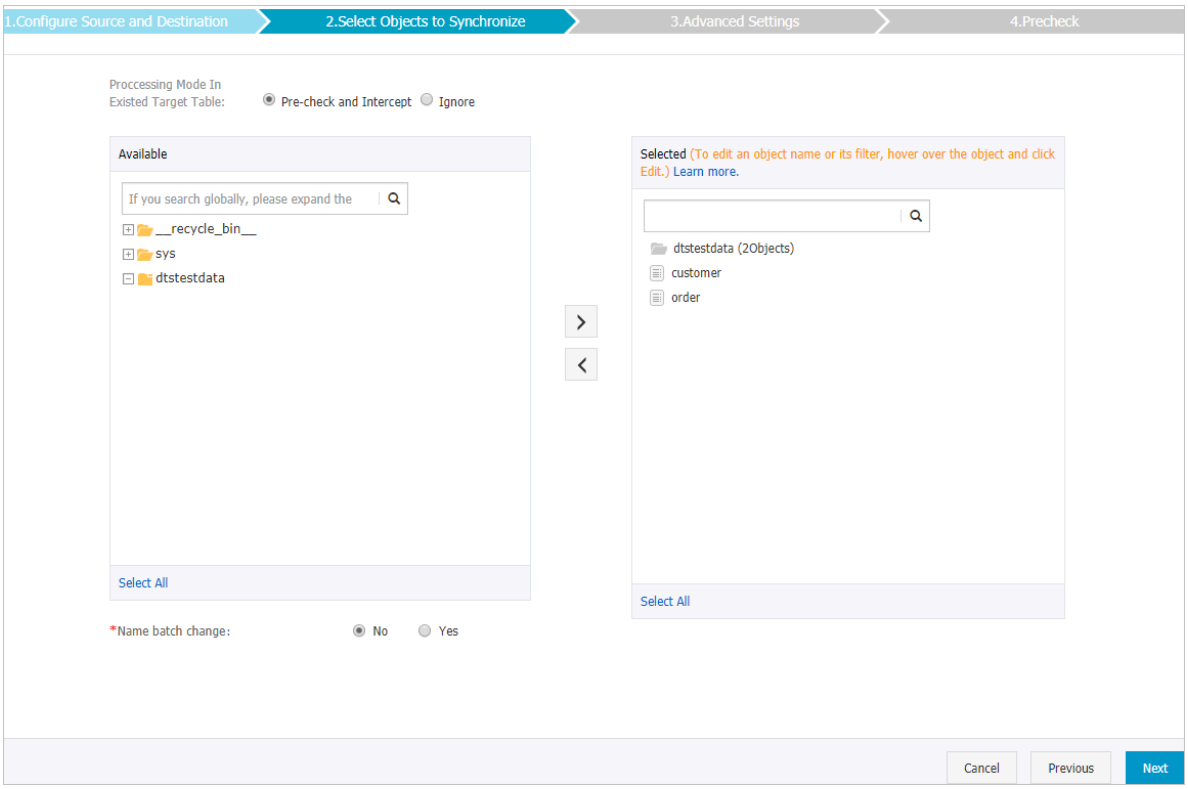
Section	Parameter	Description
Source Instance Details	Instance Type	Select **User-Created Database in ECS Instance**.
	Instance Region	The region of the source instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the 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.
	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.
	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 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.

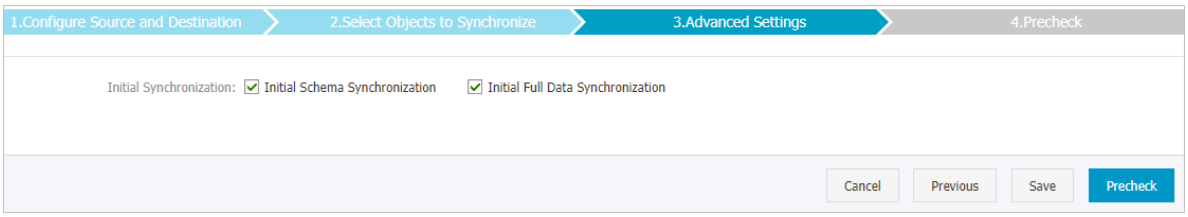
8. Configure the processing mode in existing destination tables and the objects to be synchronized.



Parameter	Description
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. - 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.

Parameter	Description
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.



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.

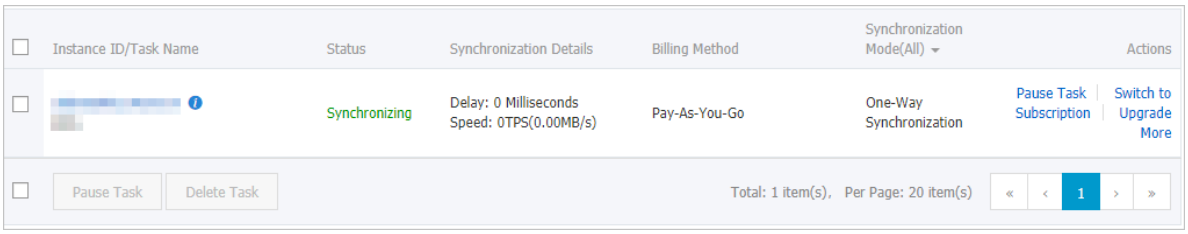
11. In the lower-right corner of the page, click **Precheck**.

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.



8.7 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 #unique_53.

Note:
DTS is allowed to access the VPC to which the user-created MySQL database belongs. For more information, see #unique_54.

- 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 gh-ost 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

Operation type	SQL statements
DML	INSERT, UPDATE, DELETE, and REPLACE

Operation 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.

Preparations

#unique_52

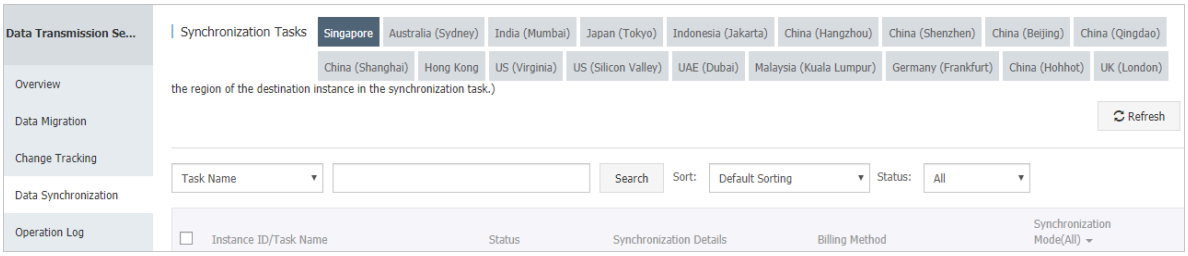
> **Note:**
> The database accounts must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.

Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

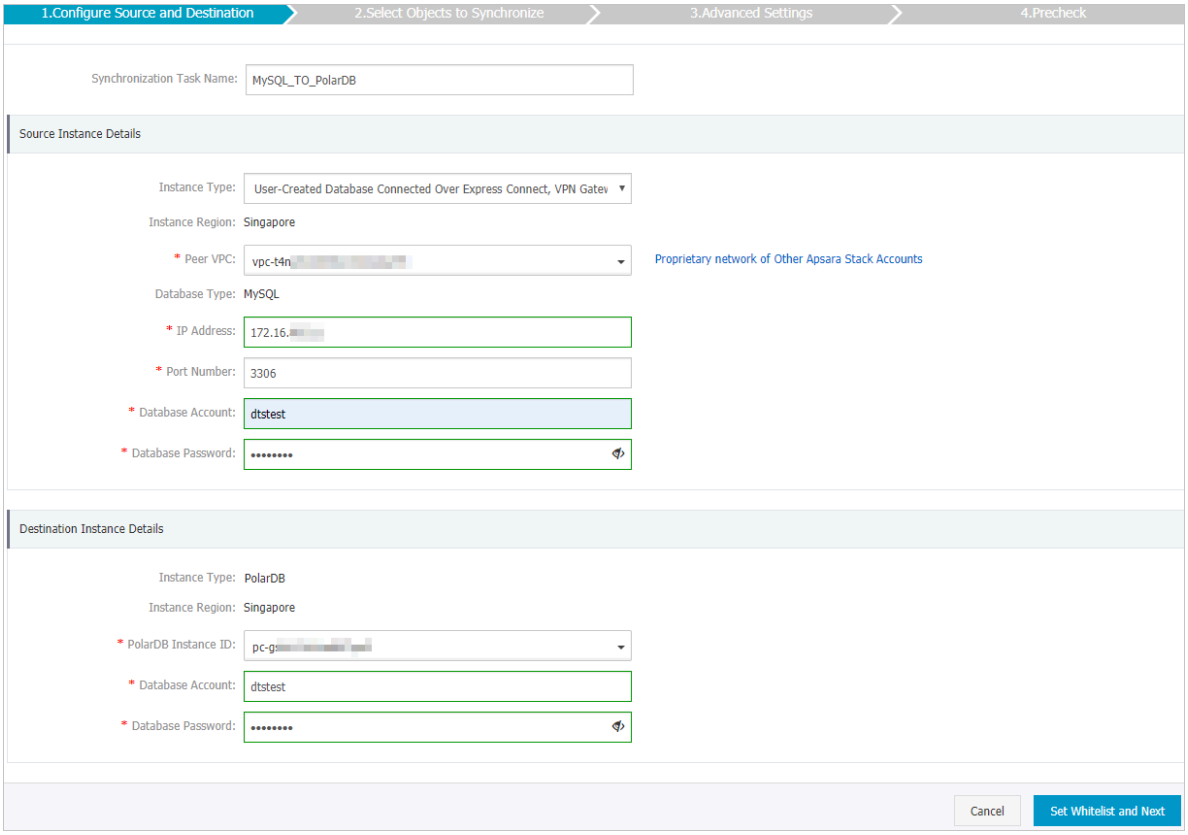
   > **Note:**
   > On the buy page, set Source Instance to **MySQL**, Target Instance to **PolarDB**, and Synchronization Topology to **One-Way Synchronization**.

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.



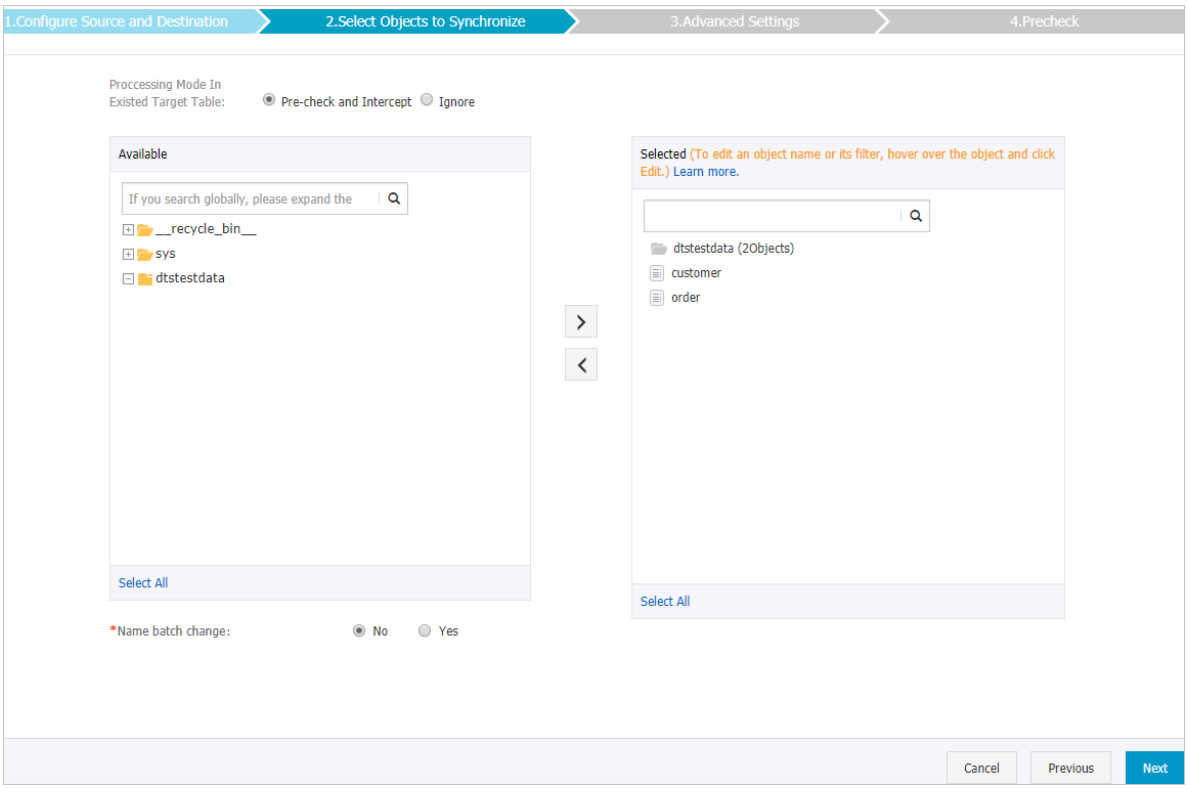
Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **User-Created Database Connected over Express 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.
	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.

Section	Parameter	Description
	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.

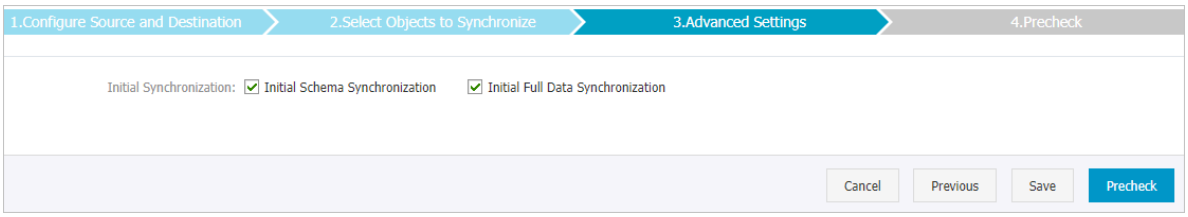
8. Configure the processing mode in existing destination tables and the objects to be synchronized.



Parameter	Description
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. - 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.

Parameter	Description
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.



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.

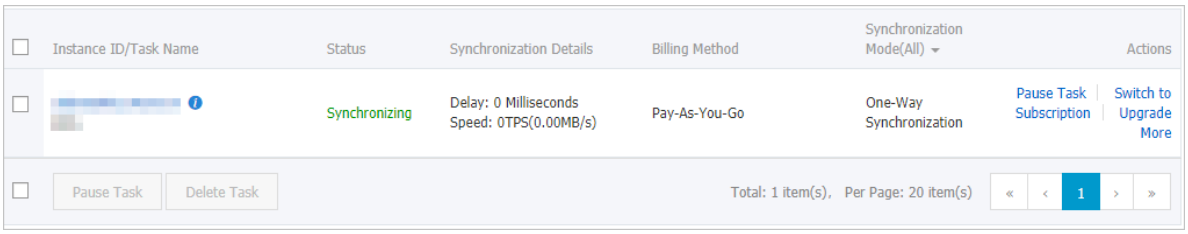
11. In the lower-right corner of the page, click **Precheck**.

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.



9 Use DTS to synchronize Redis data

9.1 Configure one-way data synchronization between ApsaraDB for Redis instances

Data Transmission Service (DTS) supports real-time one-way data synchronization between two 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 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 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. After verification, 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 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.

- If the database version of the source ApsaraDB for Redis instance is 2.8, incremental data synchronization is not supported.
- 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 noeviction, the data between the source and destination databases may become inconsistent. For more information about the data eviction policy, see #unique_62

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:

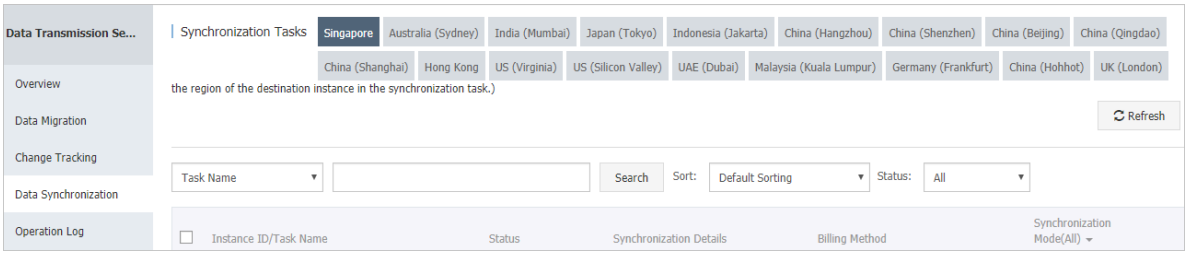
- If you use the EVAL or EVALSHA command to call Lua scripts, DTS cannot identify whether these Lua scripts are executed on the destination database. During incremental data synchronization, the destination database does not explicitly return the execution results of Lua scripts.
- When calling the SYNC 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 #unique_51/unique_51_Connect_42_section_39h_fto_gdl.

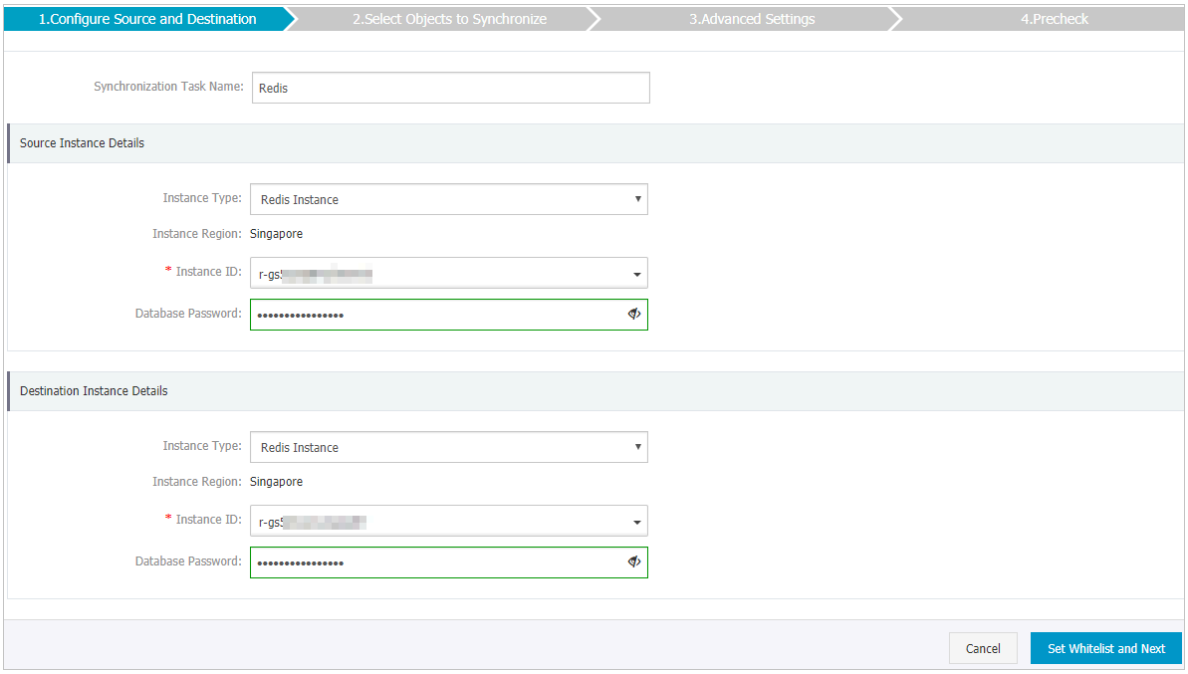
   Note:
   On the buy page, set both Source Instance and Target Instance to **Redis**.

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	Synchronization 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 **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.
	Instance ID	Select the ID of the source ApsaraDB for Redis instance.
	Database Password	Enter the database password of the source 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.

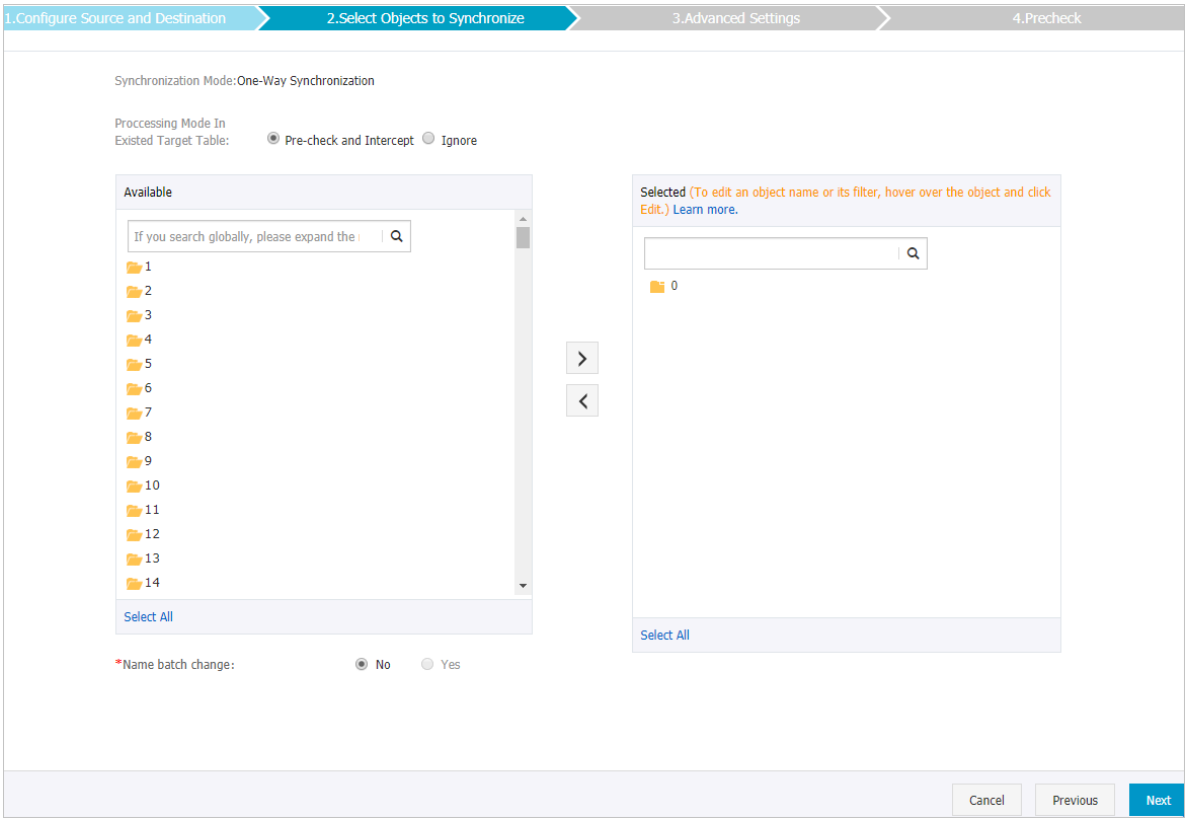
Section	Parameter	Description
Destination Instance Details	Instance Type	Select **Redis Instance**.
	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.
	Instance ID	Select the ID of the destination ApsaraDB for Redis instance.
	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.

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 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.

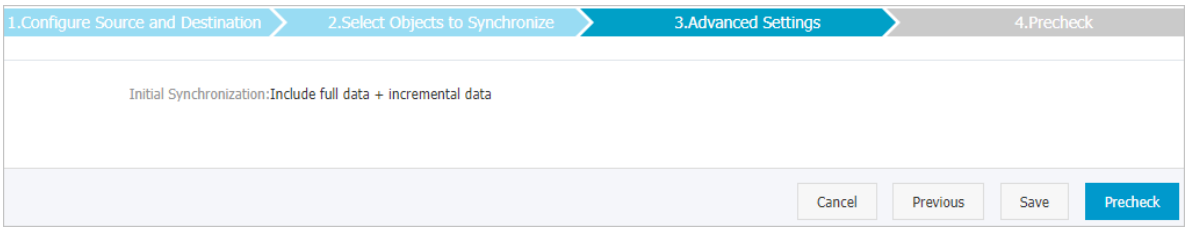


Parameter	Description
Processing Mode In Existed Target Table	• **Pre-check and Intercept**: checks whether the destination database is empty. If the destination database is empty, the precheck is passed. If the database is not empty, an error is returned during precheck and the data synchronization task cannot be started. • **Ignore**: skips the precheck for empty destination databases. **Warning:** If you select **Ignore**, the data records in the source database overwrite the data records with the same keys 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. 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.

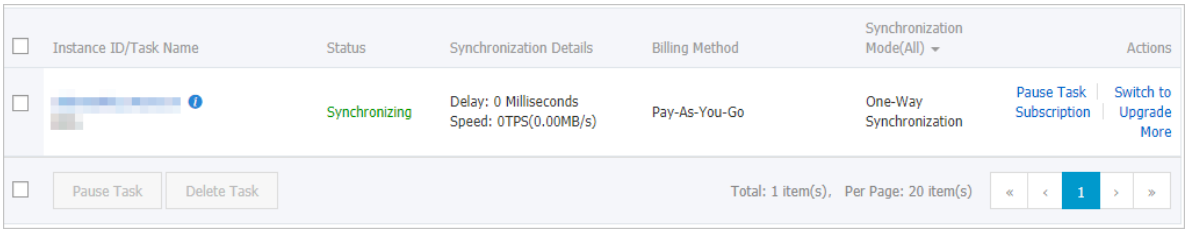


11. In the lower-right corner of the page, click **Precheck**.

> **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.



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

9.2 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:**

Prerequisites

- The version of the source Redis database is 2.8, 3.0, 3.2, 4.0, or 5.0.

   **Note:**
- 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

-
- 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.conf 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 noeviction, the data between the source and destination databases may become inconsistent. For more information about the data eviction policy, see #unique_62
-
-
-
- APPEND
-
-
-
-
-
-
-
-
-
-
-
-

Note:

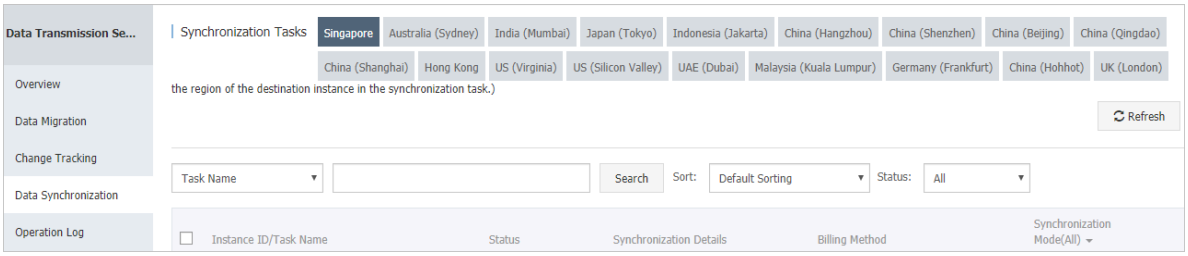
-
-

Procedure

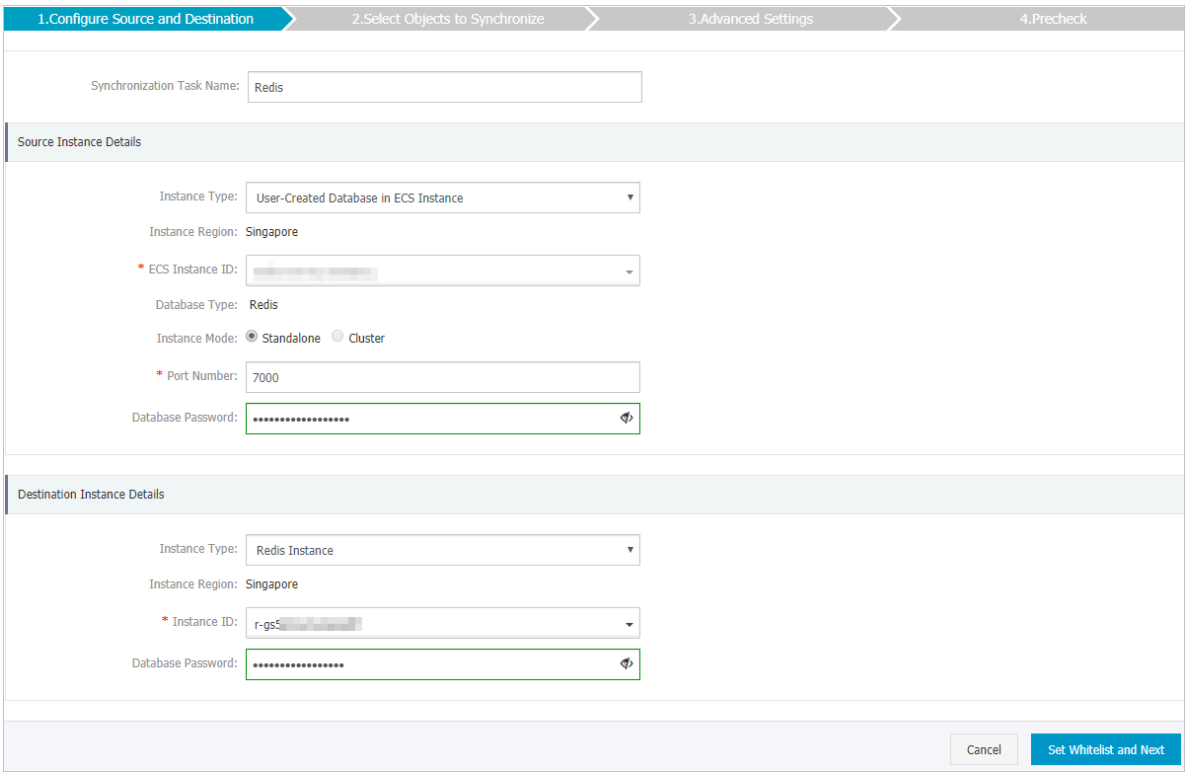
1. Purchase a data synchronization instance. For more information, see #unique_51.

   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.

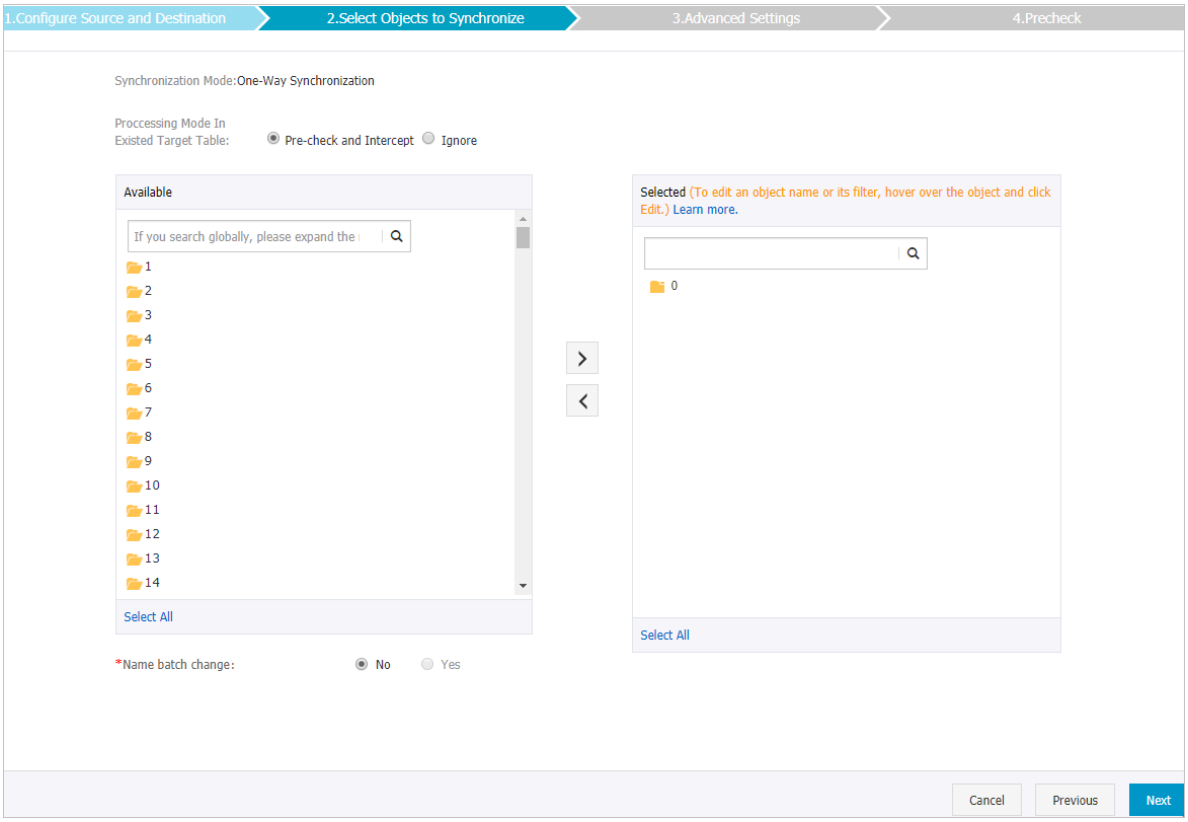


Section	Parameter	Description
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.
Source Instance Details	Instance Type	Select **User-Created Database in ECS Instance**.
	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
	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.
	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.
Destination Instance Details	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.
	Database Password	Enter the database password of the destination ApsaraDB for Redis instance.  **Note:**

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

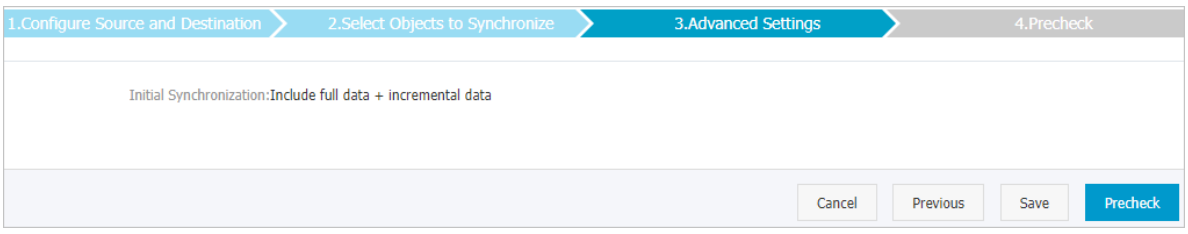
> **Note:**
> The CIDR blocks of DTS servers are automatically added to the inbound rule of the source ECS instance and the whitelist of the destination ApsaraDB for Redis instance. This ensures that DTS servers can connect to the source and destination instances.

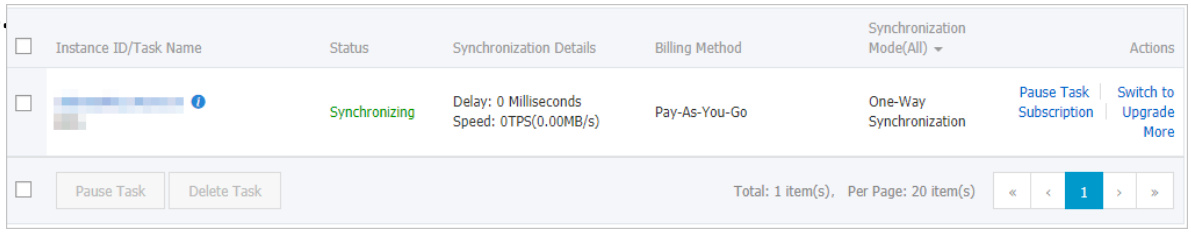
8. 

	• • **Warning:**
	• •

9.

10. > **Note:**



11. **Note:**
    -
    -
12.
13. 

    Note:

9.3 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:

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 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.
- 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.conf 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 noeviction, the data between the source and destination databases may become inconsistent. For more information about the data eviction policy, see #unique_62
-
-
-

- APPEND
-
-

-
-
-
-
-
-
-
-
-
-

Note:

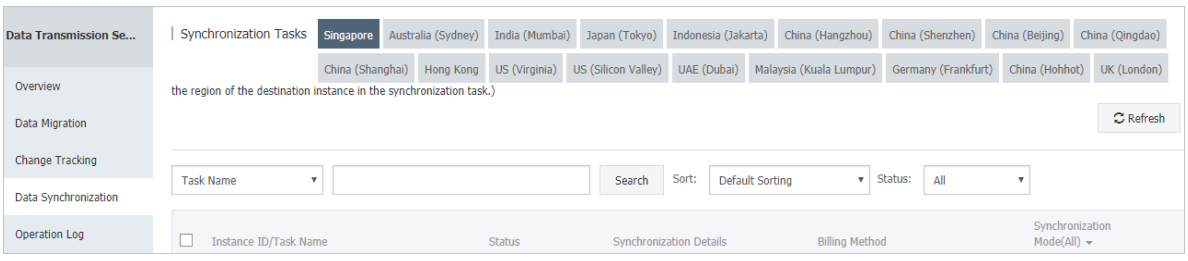
-
-

Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

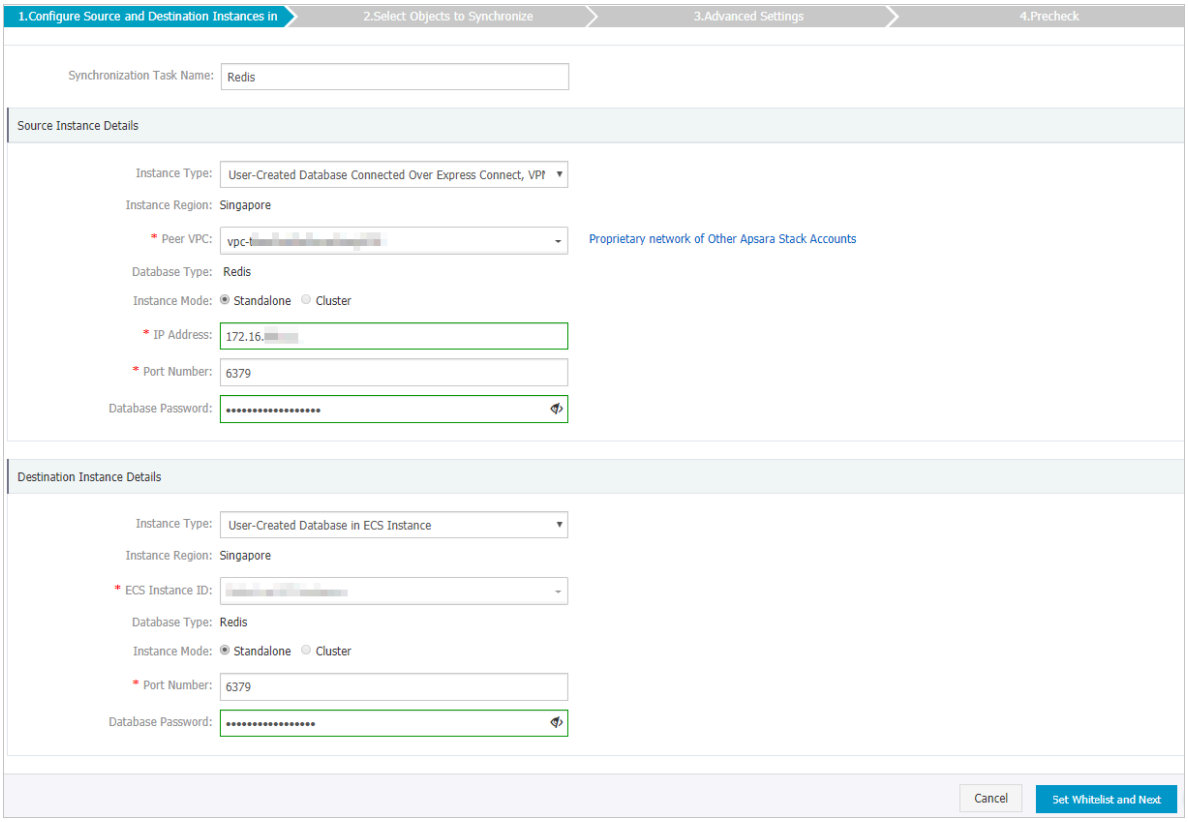
   Note:
   On the buy page, set both Source Instance and Target Instance to **Redis**.

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	Synchronization 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 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.
	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.

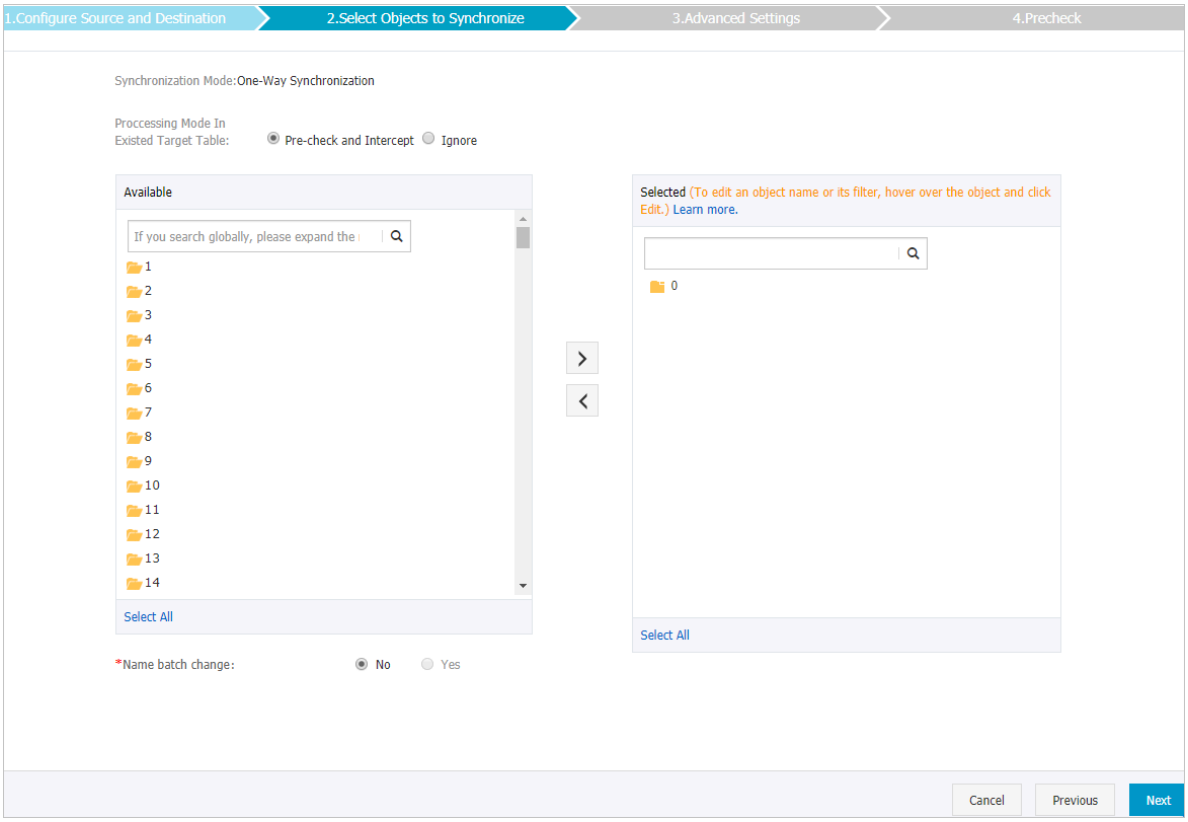
Section	Parameter	Description
	IP Address	Enter the server IP address of the source Redis database. **Note:** If the source Redis database is deployed in a cluster architecture, enter the IP address of the server to which a master node belongs.
	Port Number	Enter the service port number of the user-created Redis database. The default port number is **6379**. **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.
Destination Instance Details	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.

Section	Parameter	Description
	Port Number	Enter the service port number of the destination Redis database. The default port number is **6379**. **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. **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:

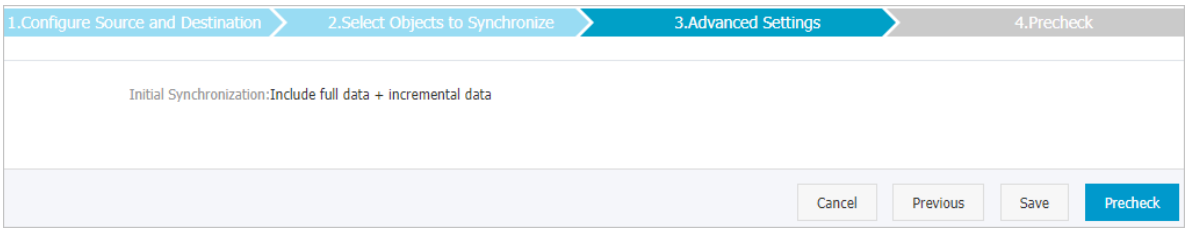
The CIDR blocks of DTS servers are automatically added to an inbound rule of the destination ECS instance. This ensures that DTS servers can connect to the destination ECS instance.

8. 

	• • **Warning:**
	• •

9.

10. **Note:**

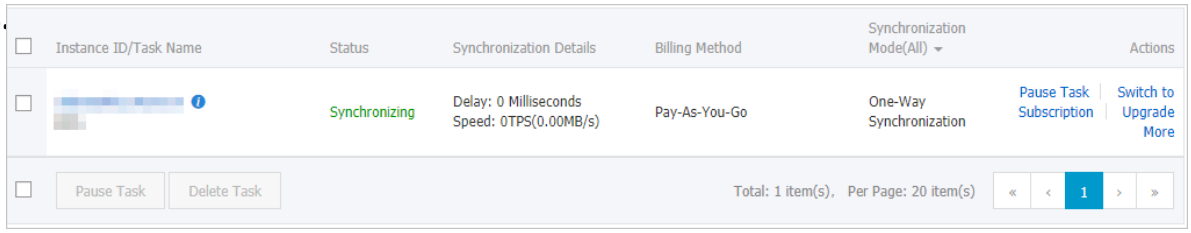


11. **Note:**

-
-

12.

13.



Note:

9.4 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 geo-redundancy, 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.conf 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 #unique_62

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.

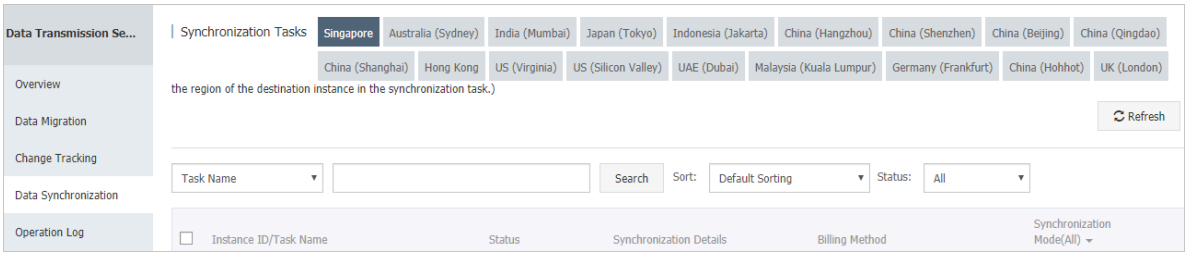
Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

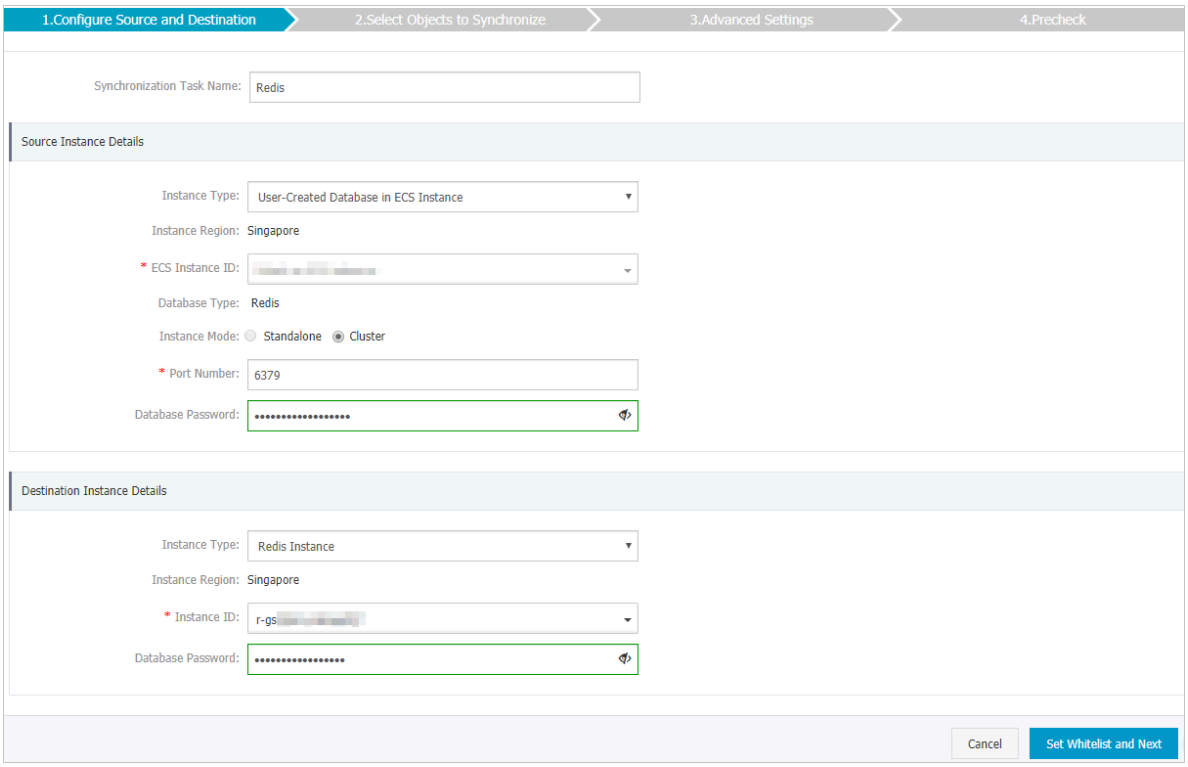
   Note:
   On the buy page, set both Source Instance and Target Instance to **Redis**.

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	Synchronization 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.

Section	Parameter	Description
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 ID	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**.
	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.
Destination Instance Details	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 cluster instance.

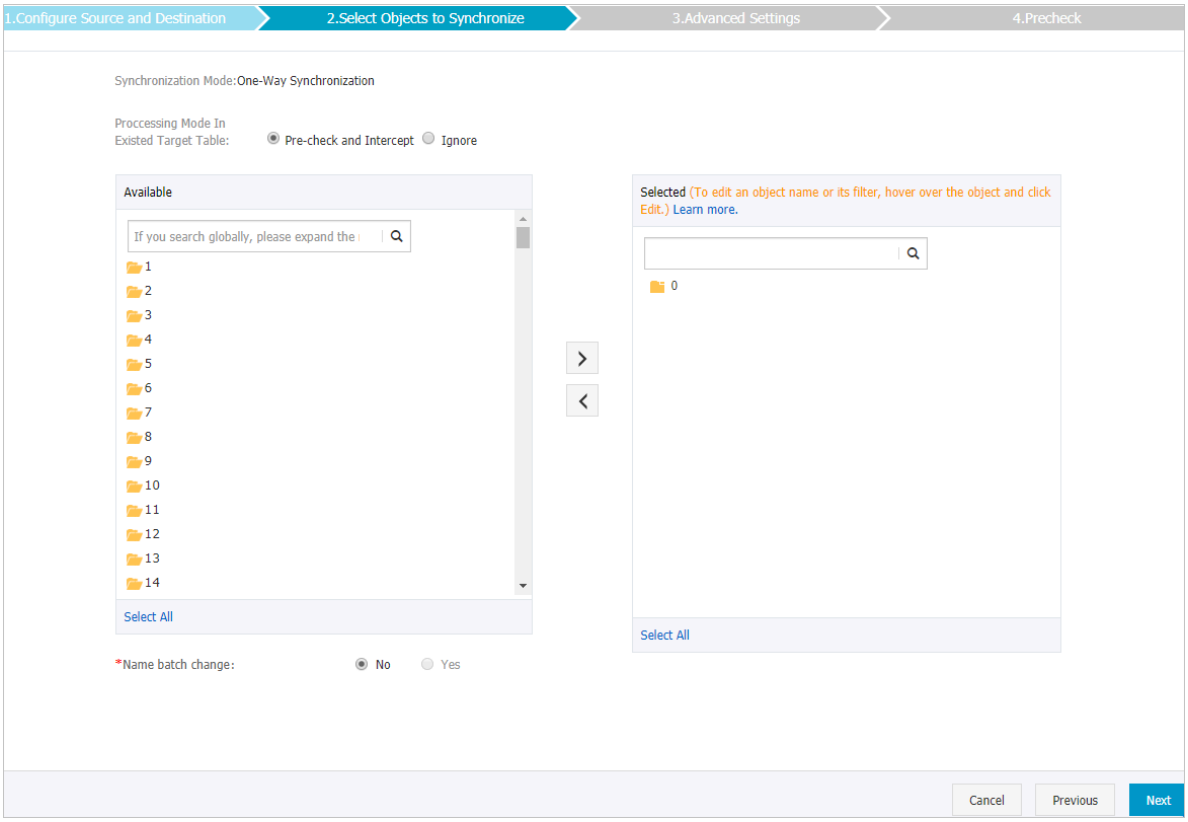
Section	Parameter	Description
	Database Password	Enter the database password of the destination ApsaraDB for Redis cluster instance.  **Note:**

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.

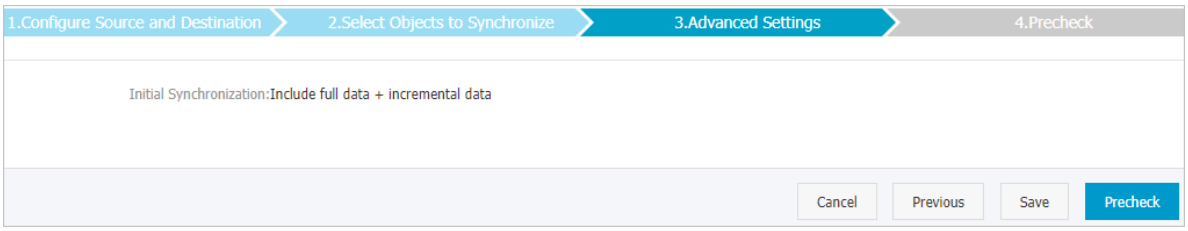


Parameter	Description
Processing Mode In Existed Target Table	• **Pre-check and Intercept**: checks whether the destination database is empty. If the destination database is empty, the precheck is passed. If the database is not empty, an error is returned during precheck and the data synchronization task cannot be started. • **Ignore**: skips the precheck for empty destination databases. **Warning:** If you select **Ignore**, the data records in the source database overwrite the data records with the same keys 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. 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.

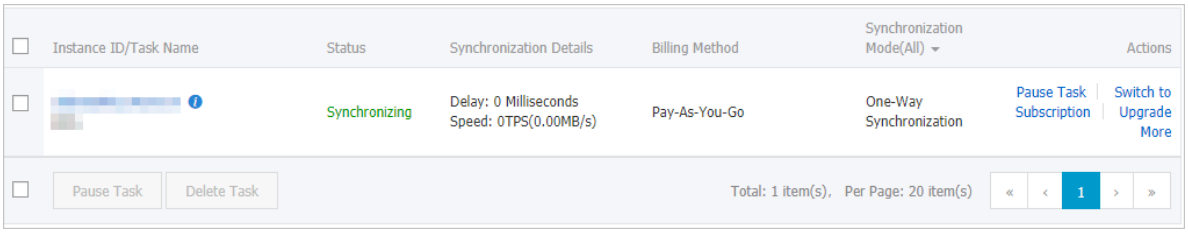


11. In the lower-right corner of the page, click **Precheck**.

> **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.



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

9.5 Synchronize data from a Codis cluster hosted on ECS to an ApsaraDB for Redis instance

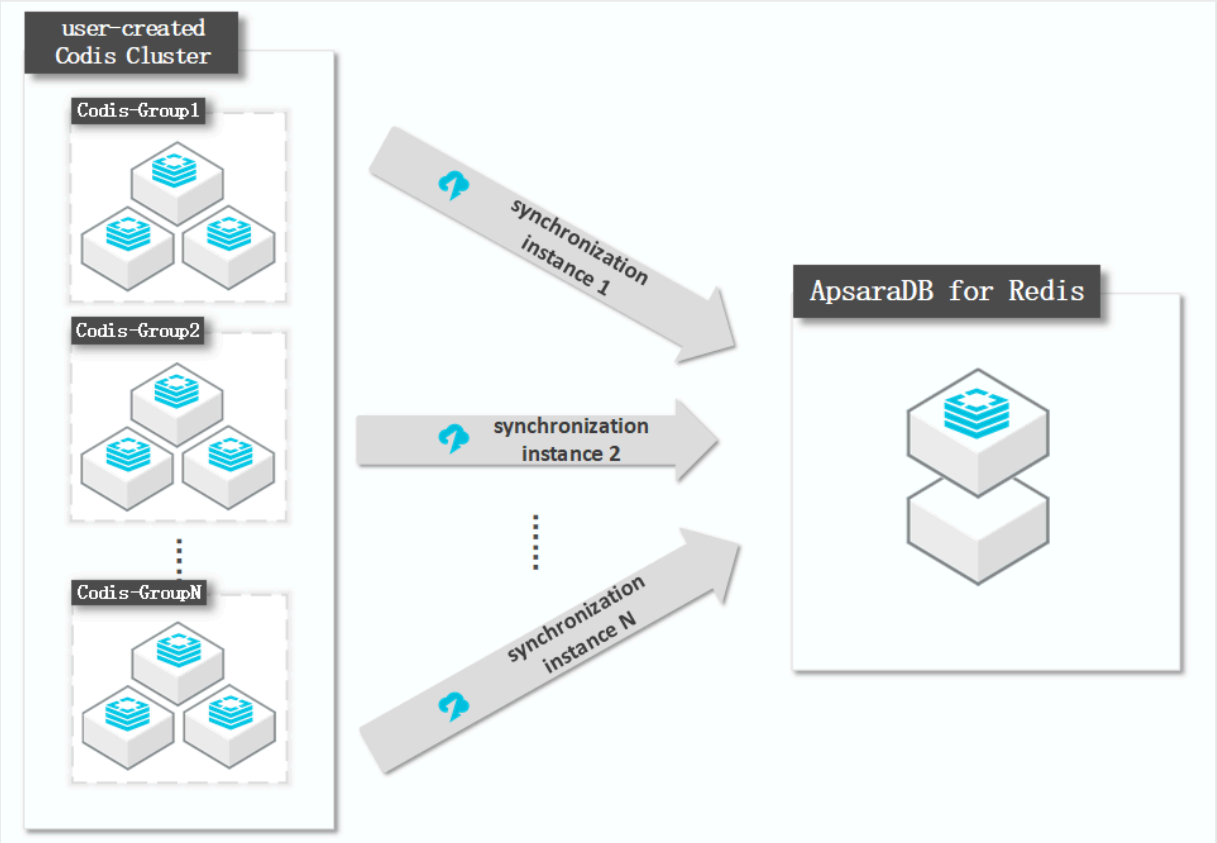
ApsaraDB for Redis is a database service compatible with the open-source Redis protocol and provides hybrid storage of memory and hard disks. Based on reliable hot standby architecture and scalable cluster architecture, ApsaraDB for Redis is suitable for scenarios that require flexible configuration 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.

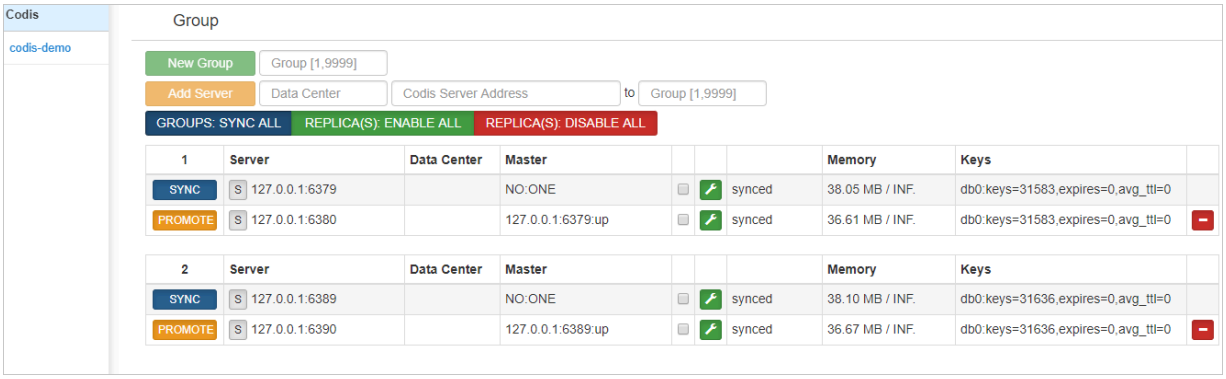
How DTS synchronizes data from a Codis cluster

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

-
- 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.conf 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 noeviction, the data between the source and destination databases may become inconsistent. For more information about the data eviction policy, see #unique_62
- 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.

-
-
-

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:**

-
-

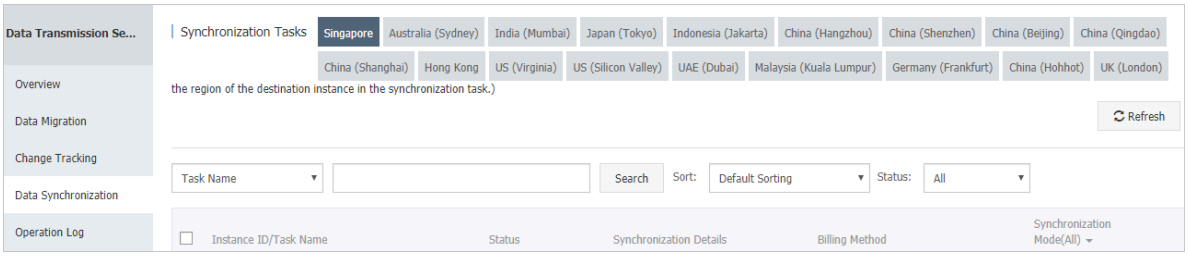
Procedure

1. Purchase a data synchronization instance. For more information, see #unique_51.

    **Note:**
   On the buy page, set Source Instance to **Redis**, Target Instance to **Redis**, and Synchronization Topology to **One-Way Synchronization**.

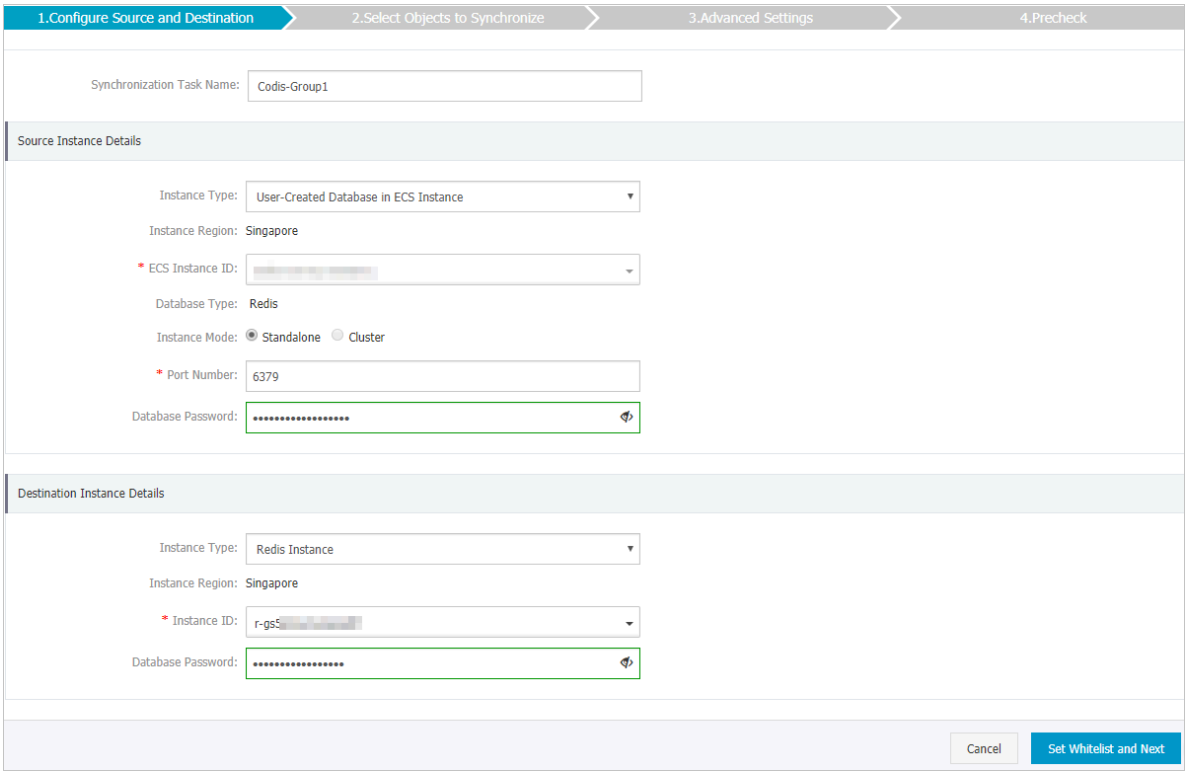
2. Log on to the DTS console.
3. In the left-side navigation pane, click **Data Synchronization**.

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	Synchronization 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**.
	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
	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.
	Database Type	The value of this parameter is set to **Redis**.
	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.
	Database Password	The database password for the master node.  **Note:** This parameter is optional and can be left blank if no database password is set.
Destination Instance Details	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.

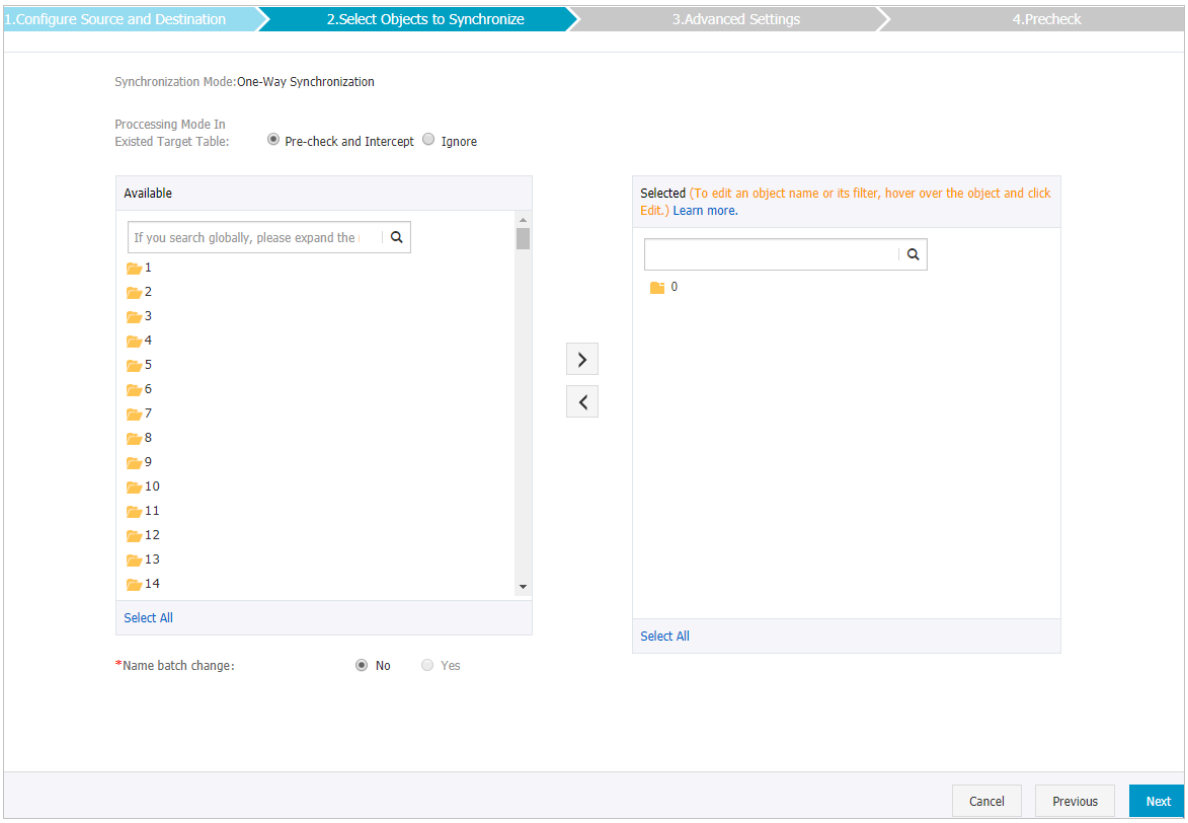
Section	Parameter	Description
	Database Password	Enter the database password of the destination ApsaraDB for Redis instance.  **Note:**

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

 **Note:**

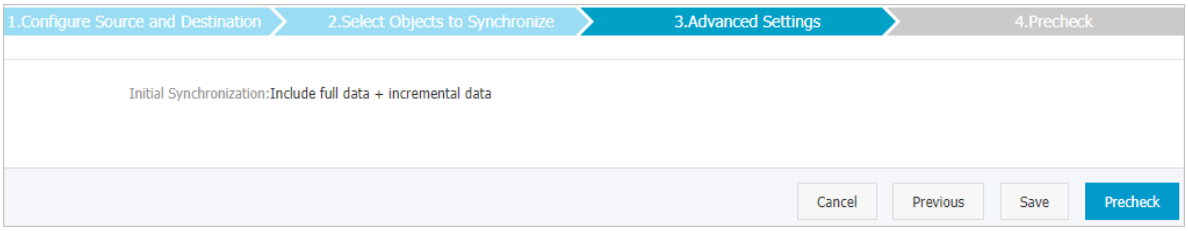
The CIDR blocks of DTS servers are automatically added to the inbound rule of the ECS instance and the whitelist of the ApsaraDB for Redis instance. This ensures that DTS servers can connect to the source and destination instances.

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 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 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.



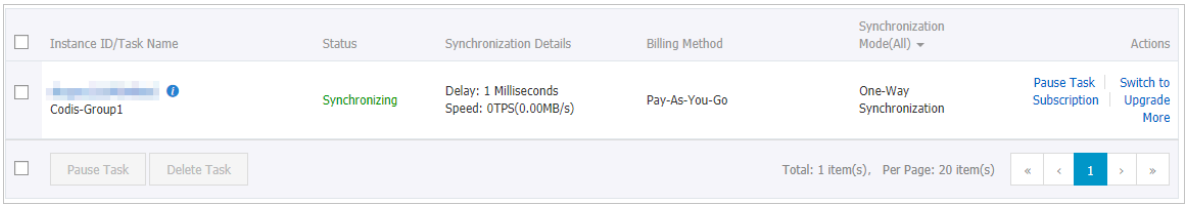
Note:

The value is set to **Include full data + incremental data**. DTS synchronizes historical data from the source Codis cluster to the destination Redis database before synchronizing incremental data.

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.

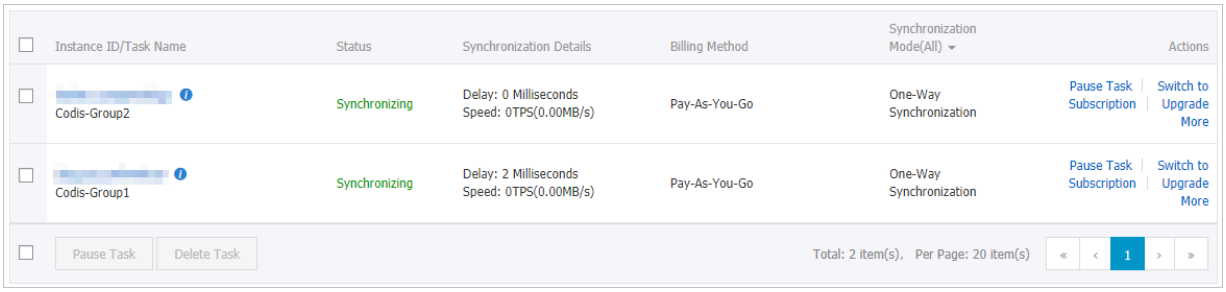


> **Note:**
>
> You can view the status of the data synchronization task on the **Synchronization Tasks** page.

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.

Figure 9-1: ApsaraDB for Redis instance

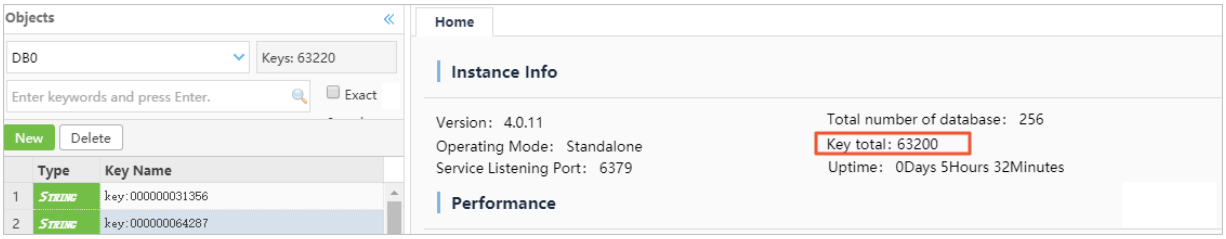
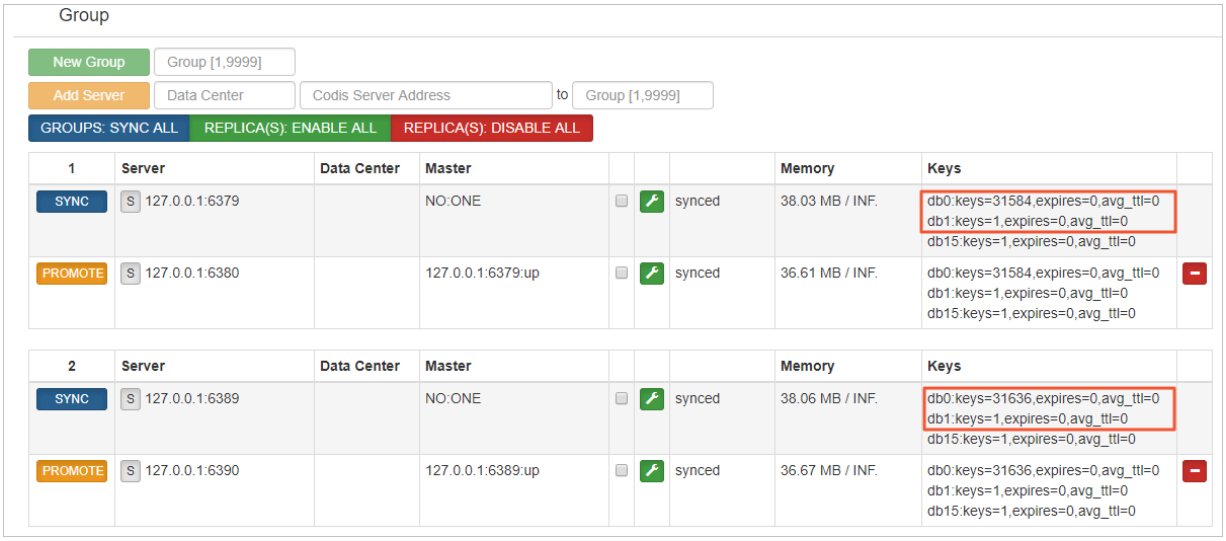


Figure 9-2: Source Codis cluster



9.6 Synchronize data from a Twemproxy Redis cluster hosted on ECS to an ApsaraDB for Redis instance

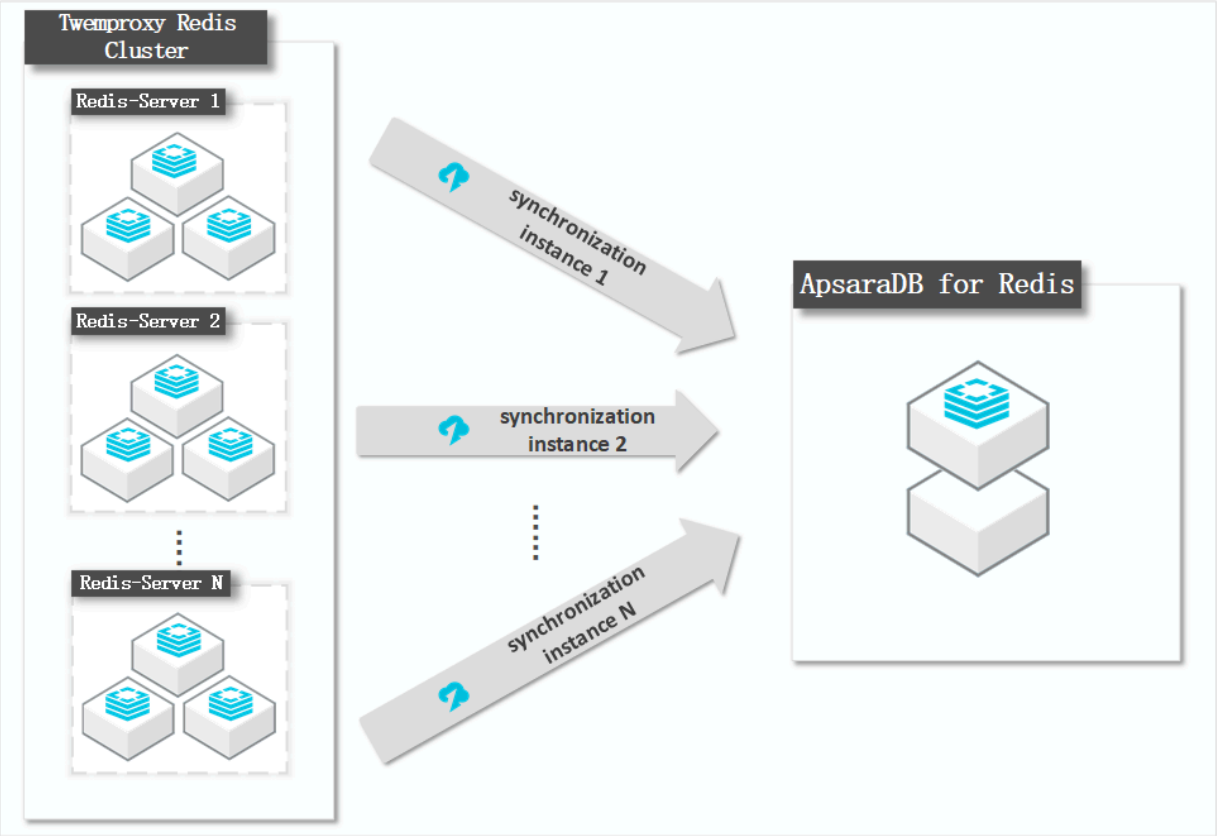
ApsaraDB for Redis is a database service compatible with the open-source Redis protocol and provides hybrid storage of memory and hard disks. Based on reliable hot standby architecture and scalable cluster architecture, ApsaraDB for Redis is suitable for scenarios that require flexible configuration 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.

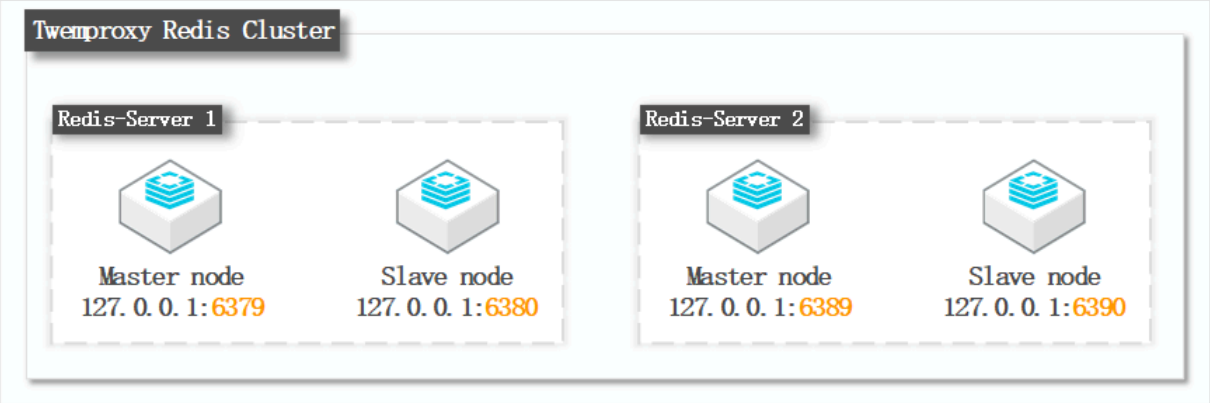
How DTS synchronizes data from a Twemproxy Redis cluster

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 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.
- 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.conf 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 noeviction, the data between the source and destination databases may become inconsistent. For more information about the data eviction policy, see #unique_62
- 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.

-
-
-

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:

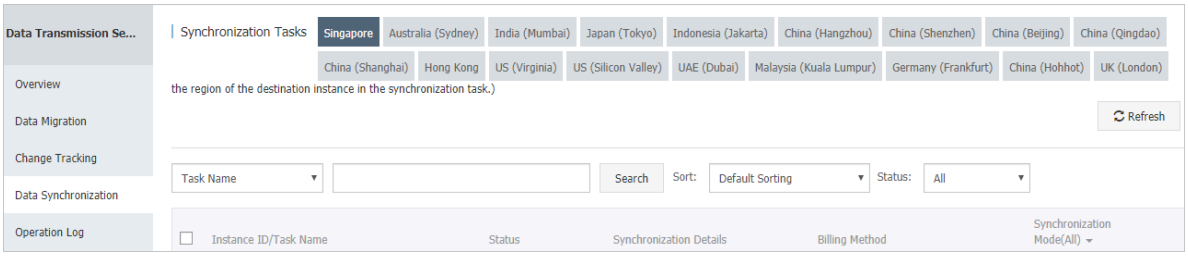
- If you use the EVAL or EVALSHA command to call Lua scripts, DTS cannot identify whether these Lua scripts are executed on the destination database. During incremental data synchronization, the destination database does not explicitly return the execution results of Lua scripts.
- When calling the SYNC 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 #unique_51.

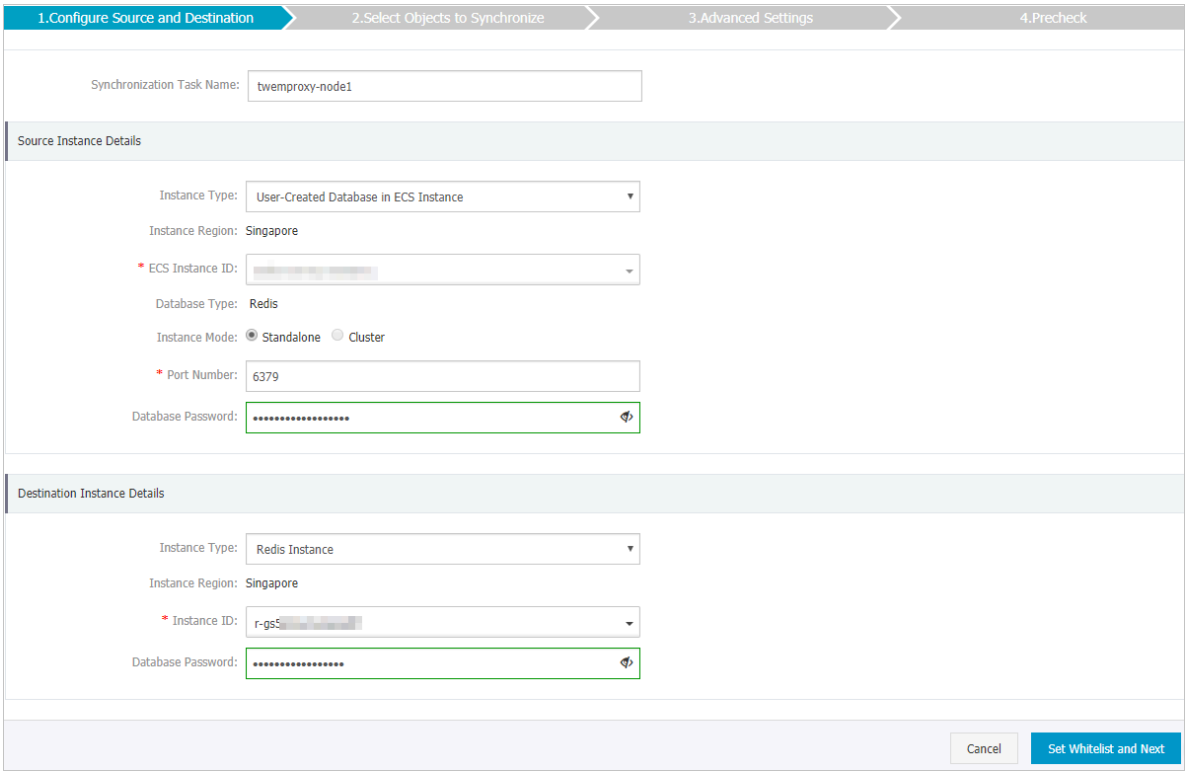
   > **Note:**
   > On the buy page, set Source Instance to **Redis**, Target Instance to **Redis**, and Synchronization Topology to **One-Way Synchronization**.

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	Synchronization 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**.
	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.

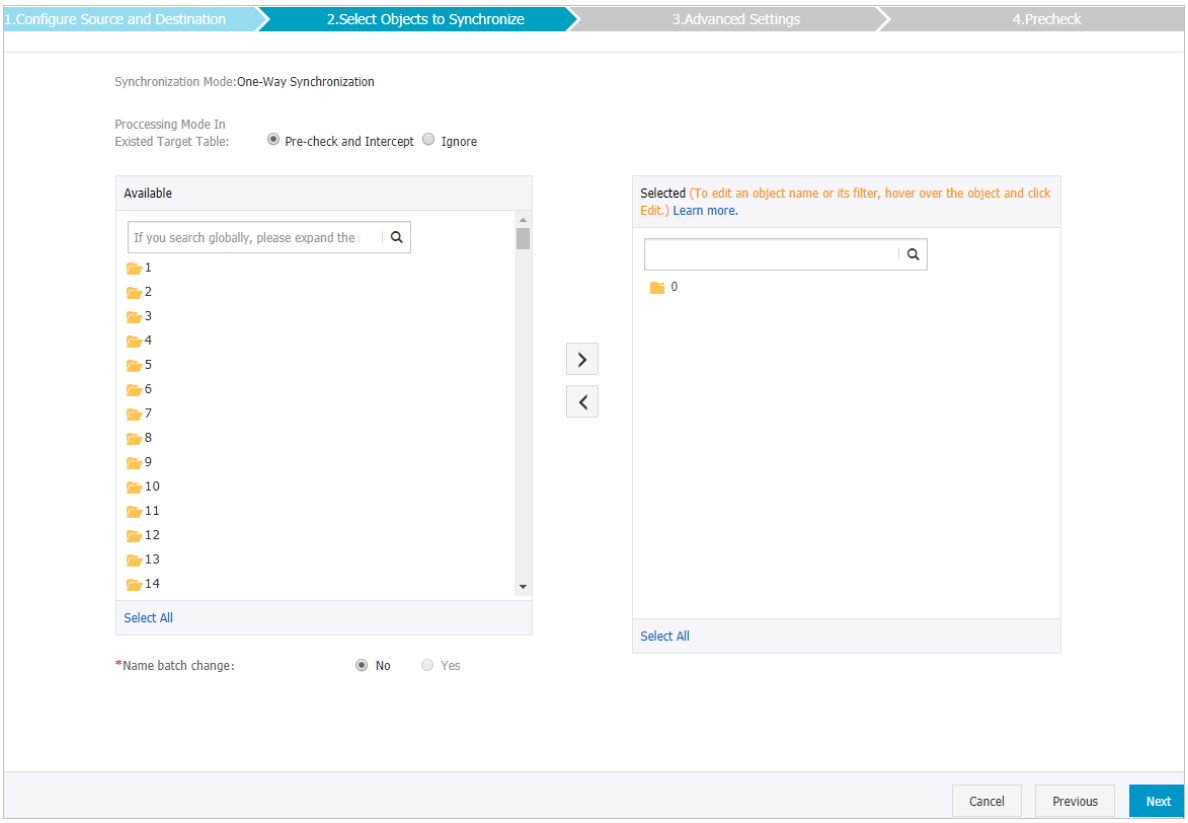
Section	Parameter	Description
	ECS Instance ID	Select the ID of the ECS instance where the master node of the Redis-Server resides.  **Note:** DTS synchronizes each Redis-Server of the Twemproxy Redis cluster by using a data synchronization task till the whole cluster is synchronized. In this step, enter 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**.
	Instance Mode	Select **Standalone**.  **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.
	Database Password	Enter the database password for the master node.  **Note:** This parameter is optional and can be left blank if no database password is set.
Destination Instance Details	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.

Section	Parameter	Description
	Database Password	Enter the database password of the destination ApsaraDB for Redis instance. **Note:**

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

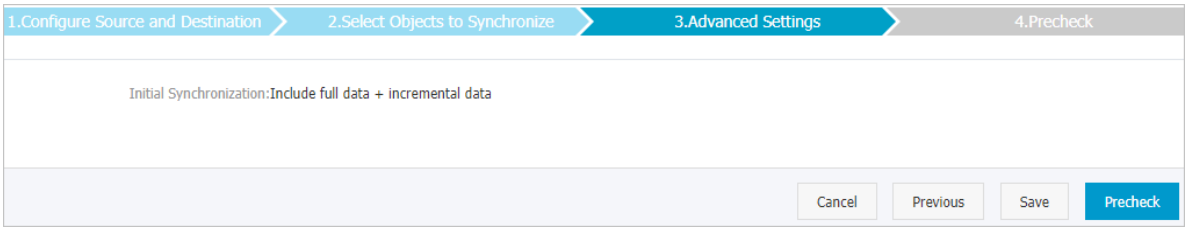
Note:
The CIDR blocks of DTS servers are automatically added to the inbound rule of the ECS instance and the whitelist of the ApsaraDB for Redis instance. This ensures that DTS servers can connect to the source and destination instances.

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 synchroniz ed. 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.



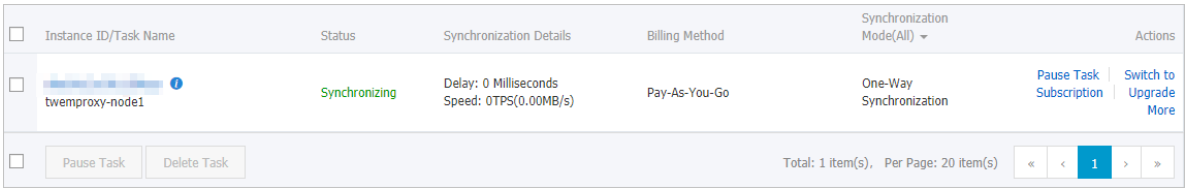
Note:

The value is set to **Include full data + incremental data**. DTS synchronizes historical data from the source Twemproxy Redis cluster to the destination Redis database before synchronizing incremental data.

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.

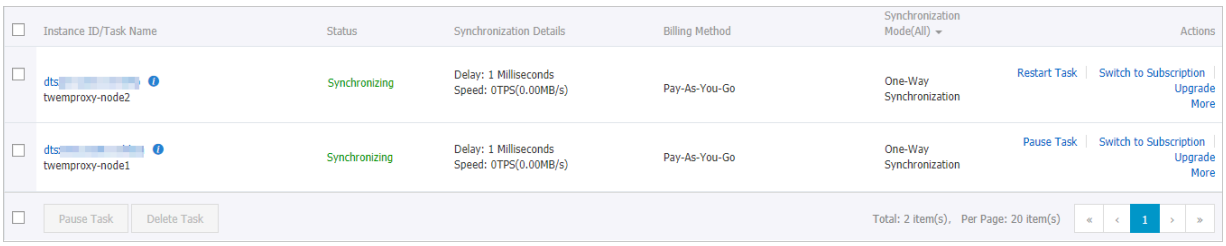


> **Note:**
>
> You can view the status of the data synchronization task on the **Synchronization Tasks** page.

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 DB0 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.

Figure 9-3: ApsaraDB for Redis instance

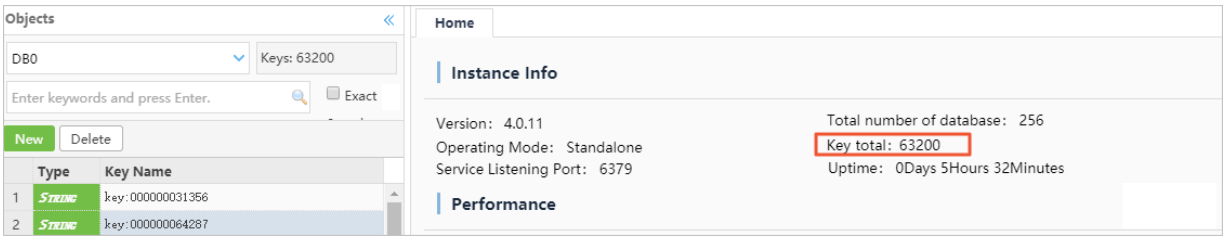


Figure 9-4: Source Twemproxy Redis cluster