# Alibaba Cloud

# Data Transmission Service Data Synchronization

Document Version: 20220712

C-J Alibaba Cloud

# Legal disclaimer

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

- You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloudauthorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
- 2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
- 3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
- 4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
- 5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud and/or its affiliates Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
- 6. Please directly contact Alibaba Cloud for any errors of this document.

# **Document conventions**

Style	Description	Example	
<u>↑</u> Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	Danger: Resetting will result in the loss of user configuration data.	
O Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.	
C) Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	Notice: If the weight is set to 0, the server no longer receives new requests.	
? Note	A note indicates supplemental instructions, best practices, tips, and other content.	Onte: 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 formatting is used for buttons ,Boldmenus, page names, and other UIelements.		Click OK.	
Courier font	Courier font is used for commands	Run the cd /d C:/window command to enter the Windows system folder.	
Italic formatting is used for parameters and variables.		bae log listinstanceid Instance_ID	
[] or [a b] This format is used for an optional value, where only one item can be selected.		ipconfig [-all -t]	
{} or {a b} This format is used for a required value, where only one item can be selected.		switch {active stand}	

# Table of Contents

1.Synchronization topologies 08
2.Data type mappings for schema synchronization
3.Precautions and limits 36
3.1. Precautions and limits for synchronizing data from a MySQ 36
3.2. Precautions and limits for synchronizing data from a Pola 64
3.3. Precautions and limits for synchronizing data from a Pola78
3.4. Limits for using PolarDB-X as the source instance 90
3.5. Solutions for using PolarDB-X as the source instance
3.6. Precautions and limits for synchronizing data from an Ora
3.7. Precautions and limits for synchronizing data from a Post 103
3.8. Precautions and limits for synchronizing data from an SQ 122
3.9. Precautions and limits for synchronizing data from a Mon 132
3.10. Precautions and limits for synchronizing data from a Db2 136
4.Prepare the database accounts for data synchronization 143
4.Prepare the database accounts for data synchronization 143 5.Synchronization task management 148
<ul> <li>4.Prepare the database accounts for data synchronization 143</li> <li>5.Synchronization task management 148</li> <li>5.1. Rename an object to be synchronized 148</li> </ul>
<ul> <li>4.Prepare the database accounts for data synchronization</li></ul>
<ul> <li>4.Prepare the database accounts for data synchronization</li></ul>
<ul> <li>4.Prepare the database accounts for data synchronization</li></ul>
4.Prepare the database accounts for data synchronization       143         5.Synchronization task management       148         5.1. Rename an object to be synchronized       148         5.2. Use SQL conditions to filter data       151         5.3. Specify the capitalization of object names in the destinati       153         5.4. Modify the naming rules for additional columns       155         5.5. View the connection status and performance of data sync       157
4.Prepare the database accounts for data synchronization1435.Synchronization task management1485.1. Rename an object to be synchronized1485.2. Use SQL conditions to filter data1515.3. Specify the capitalization of object names in the destinati1535.4. Modify the naming rules for additional columns1555.5. View the connection status and performance of data sync1575.6. Reset a data synchronization task160
4.Prepare the database accounts for data synchronization       143         5.Synchronization task management       148         5.1. Rename an object to be synchronized       148         5.2. Use SQL conditions to filter data       151         5.3. Specify the capitalization of object names in the destinati       153         5.4. Modify the naming rules for additional columns       155         5.5. View the connection status and performance of data sync       157         5.6. Reset a data synchronization task       160         5.7. Add an object to a data synchronization task       161
4.Prepare the database accounts for data synchronization1435.Synchronization task management1485.1. Rename an object to be synchronized1485.2. Use SQL conditions to filter data1515.3. Specify the capitalization of object names in the destinati1535.4. Modify the naming rules for additional columns1555.5. View the connection status and performance of data sync1575.6. Reset a data synchronization task1605.7. Add an object to a data synchronization task1615.8. Remove an object from a data synchronization task163
4.Prepare the database accounts for data synchronization       143         5.Synchronization task management       148         5.1. Rename an object to be synchronized       148         5.2. Use SQL conditions to filter data       151         5.3. Specify the capitalization of object names in the destinati       153         5.4. Modify the naming rules for additional columns       155         5.5. View the connection status and performance of data sync       157         5.6. Reset a data synchronization task       160         5.7. Add an object to a data synchronization task       161         5.8. Remove an object from a data synchronization task       163         5.9. Stop a data synchronization task       164
4.Prepare the database accounts for data synchronization       143         5.Synchronization task management       148         5.1. Rename an object to be synchronized       148         5.2. Use SQL conditions to filter data       151         5.3. Specify the capitalization of object names in the destinati       153         5.4. Modify the naming rules for additional columns       155         5.5. View the connection status and performance of data sync       157         5.6. Reset a data synchronization task       160         5.7. Add an object to a data synchronization task       161         5.8. Remove an object from a data synchronization task       163         5.9. Stop a data synchronization task       164         5.10. Troubleshoot precheck failures       165

5.12. Specify the policy for synchronizing data to Kafka partiti... 6.Synchronize data between MySQL databases ------ 176 6.1. Configure two-way data synchronization between MySQL i... ------ 176 6.2. Configure one-way data synchronization between ApsaraD... 6.3. Synchronize data from a self-managed MySQL database h...<sup>193</sup> 6.4. Synchronize data from a self-managed MySQL database c... 200 6.5. Synchronize data from an ApsaraDB RDS for MySQL insta... 207 6.6. Synchronize data between ApsaraDB RDS for MySQL insta...- 213 7.Synchronize data from a MySQL database to a different type o...... 221 7.1. Synchronize data from an ApsaraDB RDS for MySQL instan... 221 7.2. Synchronize data from a self-managed MySQL database h... 227 7.3. Synchronize data from a self-managed MySQL database co..----- 234 7.4. Synchronize data from an ApsaraDB RDS for MySQL insta... ------ 241 7.5. Synchronize data from an ApsaraDB RDS for MySQL insta... 249 7.6. Synchronize data from a self-managed MySQL database h... 256 7.7. Synchronize data from a self-managed MySQL database co...<sup>264</sup> 7.8. Synchronize data from an ApsaraDB RDS for MySQL insta... 271 7.9. Synchronize data from a self-managed MySQL database h... 280 7.10. Synchronize data from an ApsaraDB RDS for MySQL insta... 289 7.12. Synchronize data from a self-managed MySQL database t... 7.14. Synchronize data from a self-managed MySQL database h...... 323 7.15. Synchronize data from a self-managed MySQL database c... 329 8.Synchronize data to or from an Apsara PolarDB for MySQL clu...----- 342 8.1. Configure two-way data synchronization between PolarDB ...----- 342 8.2. Configure one-way data synchronization between PolarDB... 353

8.3. Synchronize data from a PolarDB for MySQL cluster to an... 8.4. Synchronize data from a PolarDB for MySQL cluster to a ... ------ 366 8.5. Synchronize data from a PolarDB for MySQL cluster to an...----- 374 8.6. Synchronize data from a PolarDB for MySQL cluster to a ... 382 8.7. Synchronize data from a PolarDB for MySQL cluster to an... 8.8. Synchronize data from a PolarDB for MySQL cluster to an... 8.9. Synchronize data from a PolarDB for MySQL cluster to a ... 403 9.Synchronize data to or from a PolarDB for Oracle cluster ------ 410 9.1. Configure one-way data synchronization between PolarDB ...----- 410 9.2. Synchronize data from a PolarDB for Oracle cluster to a s... 417 10.Synchronize data to or from a DRDS instance ------ 423 10.1. Synchronize data between PolarDB-X instances ------ 423 10.2. Synchronize data from a PolarDB-X 1.0 instance to an A... 427 10.3. Synchronize data from a PolarDB-X 1.0 instance to an A... 434 11.Synchronize data between Redis databases ------ 441 11.1. Configure one-way data synchronization between ApsaraD...- 441 11.2. Configure two-way data synchronization between Apsara... 448 11.3. Use OpenAPI Explorer to configure one-way or two-way d...... 457 11.4. Synchronize data from a self-managed Redis database h... 465 11.5. Synchronize data from a self-managed Redis database co...469 11.6. Synchronize data from a self-managed Redis cluster to a... 473 11.7. Synchronize data from a Codis cluster hosted on ECS to ... 476 11.8. Synchronize data from a Twemproxy Redis cluster hosted...-484 12.Synchronize data from an ApsaraDB RDS for SQL Server insta...... 492 12.1. Synchronize data from an ApsaraDB RDS for SQL Server ... 492 12.2. Synchronize data from a self-managed SQL Server datab... 13.Synchronize data between PostgreSQL databases ------ 513 13.1. Synchronize data from an ApsaraDB RDS for PostgreSQL .......... 513

13.2. Synchronize data from a self-managed PostgreSQL datab	520
14.Synchronize data from a TiDB database	526
14.1. Synchronize data from a self-managed TiDB database to	526

# 1.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 many-to-one synchronizatio n	Instance B Anthrophysical An	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.
	Instance A	<b>Note</b> 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 MySQL instances.
- You can upgrade the synchronization topology of a data synchronization task from oneway to two-way. For more information, see Upgrade synchronization topology from oneway to two-way.

Topology type	Topology	Description
T wo-way one- to-one synchronizatio n	two-way data synchronization MySQL MySQL	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
- Overview of data synchronization scenarios

# 2.Data type mappings for schema synchronization

Heterogeneous databases have different data types. When Data Transmission Service (DTS) synchronizes data between heterogeneous databases, DTS converts the data types of the source database to those of the destination database. This topic lists the data type mappings for you to view and evaluate the impact of data synchronization on your business.

## Overview

You can view the data type mappings between heterogeneous databases based on the following synchronization scenarios:

- Synchronize dat a from a MySQL dat abase
- Synchronize dat a from an Oracle dat abase
- Synchronize dat a from a Post greSQL dat abase
- Synchronize dat a from an SQL Server dat abase
- Synchronize dat a from a Db2 for LUW dat abase
- Synchronize data from a Db2 for i database

## Synchronize data from a MySQL database

The following table lists the data type mappings between MySQL and AnalyticDB databases. The source instance can be a self-managed MySQL database, an ApsaraDB RDS for MySQL instance, or a PolarDB for MySQL cluster. The destination instance can be an AnalyticDB for MySQL cluster V2.0 or an AnalyticDB for PostgreSQL instance.

**Note** If the value range of the data to be synchronized from the source instance exceeds the range supported by DTS, the accuracy of the data written to the destination instance will decrease.

Category	Data type of the source instance	Value range	Data type of AnalyticDB for MySQL	Data type of AnalyticDB for PostgreSQL
	BIT[(M)]	1 ~ 64	VARCHAR	BIT
	TINYINT[(M)]	-128 ~ 127	TINYINT	SMALLINT
	T INYINT [(M)] [UNSIGNED]	0 ~ 255	SMALLINT	SMALLINT
	SMALLINT[(M)]	-32768 ~ 32767	SMALLINT	SMALLINT
SMALLINT[(M)] [UNSIGNED]	0 ~ 65535	INT	INTEGER	
	MEDIUMINT[(M)]	-8388608 ~ 8388607	INT	INT EGER

#### Destination instance: an AnalyticDB for MySQL cluster or an AnalyticDB for PostgreSQL instance

Category Integer	Data type of the source instance	Value range	Data type of AnalyticDB for MySQL	Data type of AnalyticDB for PostgreSQL
	MEDIUMINT [(M)] [UNSIGNED]	0 ~ 16777215	INT	INTEGER
	INT [(M)]	-2147483648 ~ 2147483647	INT	INT EGER
	INT [(M)] [UNSIGNED]	0 ~ 4294967295	BIGINT	BIGINT
	BIGINT [(M)]	- 922337203685477 5808 ~ 922337203685477 5807	BIGINT	BIGINT
	BIGINT [(M)] [UNSIGNED]	0~ 184467440737095 51615	DECIMAL(20,0)	NUMERIC(20)
	DECIMAL[(M[,D])]	M: 0 to 65 D: 0 to 30	DECIMAL[(M[,D])]	DECIMAL
Decimal	FLOAT(p)	1.175494351E-38 ~ 3.402823466E+38	FLOAT	REAL
	DOUBLE[(M,D)]	2.22507385850720 14E-308 ~ 1.79769313486231 57E+308	DOUBLE	DOUBLE PRECISION
		1000-01-01~9999- 12-31		
	DATE	<b>Note</b> The format is YYYY-MM-DD, in UTC.	DATE	DATE
		1		

Category	Data type of the source instance	Value range	Data type of AnalyticDB for MySQL	Data type of AnalyticDB for PostgreSQL
Date and time	DAT ET IME[(fsp)]	1000-01-01 00:00:00.000000 ~ 9999-12-31 23:59:59.9999999 7 Note The format is YYYY-MM-DD hh:mm:ss[.fr action], in UT C.	DAT ET IME	T IMEST AMP
	TIMESTAMP[(fsp)]	1970-01-01 00:00:01.000000 ~ 2038-01-19 03:14:07.9999999	TIMESTAMP	T IMEST AMP WIT H T IME ZONE
	TIME[(fsp)] YEAR[(4)]	- 838:59:59.000000 ~ 838:59:59.000000 838:59:59.000000 <b>?</b> Note The format is hh:mm:ss[.fr action], in UT C. 1901 to 2155, or	TIME	T IME INT EGER
	CHAR[(M)]	0 to 255 characters	VARCHAR	CHAR
	VARCHAR(M)	0 to 65,535 characters	VARCHAR	VARCHAR

Category	Data type of the source instance	Value range	Data type of AnalyticDB for MySQL	Data type of AnalyticDB for PostgreSQL
	BINARY[(M)]	0 to 255 bytes	VARBINARY	BYTEA
	VARBINARY(M)	0 to 65,535 bytes	VARBINARY	BYTEA
	TINYBLOB	255 (2^8 - 1) bytes	VARBINARY	BYTEA
	TINYTEXT	255 (2^8 - 1) characters	VARCHAR	TEXT
	BLOB	65,535 (2^16 - 1) bytes	VARBINARY	BYTEA
String	TEXT	65,535 (2^16 - 1) characters	VARCHAR	TEXT
	MEDIUMBLOB	16,777,215 (2^24 - 1) bytes	VARBINARY	BYTEA
	MEDIUMT EXT	16,777,215 (2^24 - 1) characters	VARCHAR	TEXT
	LONGBLOB	4,294,967,295 or 4 GB (2^32 - 1) bytes	VARBINARY	BYTEA
	LONGTEXT	4,294,967,295 or 4 GB (2^32 - 1) characters	VARCHAR	TEXT
	ENUM('value1','val ue2',)	An ENUM column can have a maximum of 65,535 distinct elements.	VARCHAR	VARCHAR(128)
	SET ('value1','value 2',)	A SET column can have a maximum of 64 distinct elements.	VARCHAR	VARCHAR(128)
	GEOMET RY	Geometry values of any type	VARBINARY	POLYGON
	POINT	N/A	VARBINARY	POINT
	LINESTRING	N/A	VARBINARY	PATH
	POLYGON	N/A	VARBINARY	POLYGON
	MULTIPOINT	N/A	VARBINARY	POLYGON

Spatial Category	Data type of the source instance	Value range	Data type of AnalyticDB for MySQL	Data type of AnalyticDB for PostgreSQL
	MULT ILINEST RING	N/A	VARBINARY	PATH
	MULTIPOLYGON	N/A	VARBINARY	POLYGON
	GEOMET RY COLLEC T ION	A collection of geometry values of any type	VARBINARY	POLYGON
JSON	JSON	N/A	JSON	JSON

#### Destination instance: a DataHub project, a Message Queue for Apache Kafka instance, or a selfmanaged Kafka cluster

Category	Data type of the source instance	Value range	Data type of DataHub	Data type of a Message Queue for Apache Kafka instance or a self- managed Kafka cluster
	BIT [(M)]	1 ~ 64	BOOLEAN   STRING	
	TINYINT[(M)]	-128 ~ 127	BIGINT	
	T INYINT [(M)] [UNSIGNED]	0 ~ 255	BIGINT	
	SMALLINT[(M)]	-32768 ~ 32767	BIGINT	
	SMALLINT[(M)] [UNSIGNED]	0 ~ 65535	BIGINT	
	MEDIUMINT [(M)]	-8388608 ~ 8388607	BIGINT	
Integer	MEDIUMINT [(M)] [UNSIGNED]	0 ~ 16777215	BIGINT	
	INT[(M)]	-2147483648 ~ 2147483647	BIGINT	
	INT [(M)] [UNSIGNED]	0 ~ 4294967295	BIGINT	

Category	Data type of the source instance	Value range	Data type of DataHub	Data type of a Message Queue for Apache Kafka instance or a self- managed Kafka cluster
	BIGINT [(M)]	- 922337203685477 5808 ~ 922337203685477 5807	BIGINT	
	BIGINT [(M)] [UNSIGNED]	0 ~ 184467440737095 51615	BIGINT	
	DECIMAL[(M[,D])]	M: 0 to 65 D: 0 to 30	DECIMAL	
Decimal	FLOAT (p)	1.175494351E-38 ~ 3.402823466E+38	DOUBLE	
	DOUBLE[(M,D)]	2.22507385850720 14E-308 ~ 1.79769313486231 57E+308	DOUBLE	
		1000-01-01~9999- 12-31		
	DATE	<b>Note</b> The format is YYYY-MM-DD, in UTC.	TIMESTAMP	
		1000-01-01 00:00:00.000000 ~ 9999-12-31 23:59:59.9999999		
	DAT ET IME[(fsp)]	<b>Note</b> The format is YYYY-MM-DD hh:mm:ss[.fr action], in UTC.	TIMESTAMP	
		1	,	

Category Date and time	Data type of the source instance	Value range	Data type of DataHub	Data type of a Message Queue for Apache Kafka instance or a self- managed Kafka cluster
		1970-01-01 00:00:01.000000 ~ 2038-01-19 03:14:07.999999		Consistant with
	TIMEST AMP[(fsp)]		TIMESTAMP	the data types of MySQL or PolarDB for MySQL
	TIME[(fsp)]	- 838:59:59.000000 ~ 838:59:59.000000 7 838:59:59.000000 7 838:59:59.000000 7 838:59:59.000000 7 838:59:59.000000 7 838:59:59.000000	STRING	
	YEAR[(4)]	1901 to 2155, or 0000	STRING	
	CHAR[(M)]	0 to 255 characters	STRING	
	VARCHAR(M)	0 to 65,535 characters	STRING	-
	BINARY[(M)]	0 to 255 bytes	STRING	
	VARBINARY(M)	0 to 65,535 bytes	STRING	
	TINYBLOB	255 (2^8 - 1) bytes	STRING	
	TINYTEXT	255 (2^8 - 1) characters	STRING	
	BLOB	65,535 (2^16 - 1) bytes	STRING	

Category	Data type of the source instance	Value range	Data type of DataHub	Data type of a Message Queue for Apache Kafka instance or a self- managed Kafka cluster
String	TEXT	65,535 (2^16 - 1) characters	STRING	
	MEDIUMBLOB	16,777,215 (2^24 - 1) bytes	STRING	
	MEDIUMT EXT	16,777,215 (2^24 - 1) characters	STRING	
	LONGBLOB	4,294,967,295 or 4 GB (2^32 - 1) bytes	STRING	
	LONGTEXT	4,294,967,295 or 4 GB (2^32 - 1) characters	STRING	
	ENUM('value1','val ue2',)	An ENUM column can have a maximum of 65,535 distinct elements.	STRING	
	SET('value1','value 2',)	A SET column can have a maximum of 64 distinct elements.	STRING	
	GEOMET RY	Geometry values of any type	STRING	
	POINT	N/A	STRING	
Spatial	LINESTRING	N/A	STRING	
	POLYGON	N/A	STRING	
	MULTIPOINT	N/A	STRING	
	MULT ILINEST RING	N/A	STRING	
	MULTIPOLYGON	N/A	STRING	
	GEOMET RY COLLEC T ION	A collection of geometry values of any type	STRING	
JSON	JSON	N/A	STRING	

Category	Data type of the source instance	Value range	MaxCompute	Elasticsearch
	BIT[(M)]	1 ~ 64	BOOLEAN   STRING	BOOLEAN   LONG Note If the data is only one byte long, we recommend that you use the BOOLEAN data type in Elasticsearch.
	TINYINT[(M)]	-128 ~ 127	BIGINT	SHORT
	T INYINT [(M)] [UNSIGNED]	0 ~ 255	BIGINT	INTEGER
Integer	SMALLINT[(M)]	-32768 ~ 32767	BIGINT	SHORT
	SMALLINT[(M)] [UNSIGNED]	0 ~ 65535	BIGINT	INTEGER
	MEDIUMINT [(M)]	-8388608 ~ 8388607	BIGINT	INTEGER
	MEDIUMINT [(M)] [UNSIGNED]	0 ~ 16777215	BIGINT	INTEGER
	INT[(M)]	-2147483648 ~ 2147483647	BIGINT	INTEGER
	INT [(M)] [UNSIGNED]	0 ~ 4294967295	BIGINT	LONG
	BIGINT [(M)]	- 922337203685477 5808 ~ 922337203685477 5807	BIGINT	LONG
	BIGINT [(M)] [UNSIGNED]	0 ~ 184467440737095 51615	BIGINT	LONG

#### Destination instance: a MaxCompute project or an Elasticsearch cluster

Category	Data type of the source instance	Value range	MaxCompute	Elasticsearch
Decimal Decimal Decimal Decimal Decimal Decimal	DECIMAL[(M[,D])]	M: 0 to 65 D: 0 to 30	DOUBLE	DOUBLE Note If the DECIMAL value contains a decimal point, we recommend that you use the TEXT data type in Elasticsearch to ensure data consistency.
	FLOAT (p)	1.175494351E-38 ~ 3.402823466E+38	DOUBLE	FLOAT
	DOUBLE[(M,D)]	2.22507385850720 14E-308 ~ 1.79769313486231 57E+308	DOUBLE	DOUBLE
	DATE	1000-01-01~9999- 12-31 ? Note         The format is         YYYY-MM-DD,         in UTC.	DAT ET IME	DATE Note The format is YYYY-MM-DD, in UT C. For more information, see Date format mappings.

Category	Data type of the source instance	Value range	MaxCompute	Elasticsearch
	DAT ET IME[(fsp)]	1000-01-01 00:00:0000000 ~ 9999-12-31 23:59:59.9999999 ⑦ Note       The format is       YYYY-MM-DD       hh:mm:ss[.fr       action], in       UT C.	DAT ET IME	DATE Note 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.
Date and time	TIMEST AMP[(fsp)]	1970-01-01 00:00:01.000000 ~ 2038-01-19 03:14:07.9999999	DAT ET IME	DATE <b>?</b> Note 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.

Category	Data type of the source instance	Value range	MaxCompute	Elasticsearch
T II YE	TIME[(fsp)]	- 838:59:59.000000 ~ 838:59:59.000000 ? Note The format is hh:mm:ss[.fr action], in UT C.	STRING	DATE <b>?</b> Note The DATE format is YYYY-MM-DD, in UTC. For more information, see Date format mappings.
	YEAR[(4)]	1901 to 2155, or 0000	STRING	DATE ONOTE         The DATE         format is         yyyy, in UTC.         For more         information,         see Date         format         mappings.
	CHAR[(M)]	0 to 255 characters	STRING	TEXT
	VARCHAR(M)	0 to 65,535 characters	STRING	TEXT
	BINARY[(M)]	0 to 255 bytes	STRING	BINARY
	VARBINARY(M)	0 to 65,535 bytes	STRING	BINARY
T INYBLC T INYTEX BLOB TEXT	TINYBLOB	255 (2^8 - 1) bytes	STRING	BINARY
	TINYTEXT	255 (2^8 - 1) characters	STRING	TEXT
	BLOB	65,535 (2^16 - 1) bytes	STRING	BINARY
	TEXT	65,535 (2^16 - 1) characters	STRING	TEXT
	MEDIUMBLOB	16,777,215 (2^24 - 1) bytes	STRING	BINARY

र्दभांबgory	Data type of the source instance	Value range	MaxCompute	Elasticsearch
	MEDIUMTEXT	16,777,215 (2^24 - 1) characters	STRING	TEXT
	LONGBLOB	4,294,967,295 or 4 GB (2^32 - 1) bytes	STRING	BINARY
	LONGTEXT	4,294,967,295 or 4 GB (2^32 - 1) characters	STRING	ТЕХТ
	ENUM('value1','val ue2',)	An ENUM column can have a maximum of 65,535 distinct elements.	STRING	KEYWORD
	SET ('value1','value 2',)	A SET column can have a maximum of 64 distinct elements.	STRING	KEYWORD
	GEOMET RY	Geometry values of any type	STRING	GEO_SHAPE
	POINT	N/A	STRING	GEO_POINT
	LINESTRING	N/A	STRING	GEO_SHAPE
	POLYGON	N/A	STRING	GEO_SHAPE
				GEO_SHAPE
Spatial	MULT IPOINT	N/A	STRING	<b>Note</b> If the data is only one byte long, we recommend that you use the BOOLEAN data type in Elasticsearch.
	MULT ILINEST RING	N/A	STRING	GEO_SHAPE
	MULTIPOLYGON	N/A	STRING	GEO_SHAPE

Category	Data type of the source instance	Value range	MaxCompute	Elasticsearch
	GEOMET RY COLLEC T ION	A collection of geometry values of any type	STRING	GEO_SHAPE
JSON	JSON	N/A	STRING	OBJECT Note If the data is only one byte long, we recommend that you use the BOOLEAN data type in Elasticsearch.

## Synchronize data from an Oracle database

The following table lists the data type mappings between a self-managed Oracle database and an AnalyticDB for PostgreSQL instance.

**Note** If the value range of the data to be synchronized from the source instance exceeds the range supported by DTS, the accuracy of the data written to the destination instance will decrease.

Category	Data type of Oracle	Value range	Data type of AnalyticDB for PostgreSQL
Numeric	NUMBER(p,s)	1 to 22 bytes The argument p indicates the precision. Valid values: 1 to 38. The argument s indicates the scale. Valid values: -84 to 127.	DECIMAL   TINYINT   SMALLINT   INTEGER   BIGINT
	FLOAT(p)	1 to 22 bytes The variable p indicates a pointer. Valid values: 1 to 126 bits.	DOUBLE PRECISION
	BINARY_FLOAT	A 32-bit floating-point number (4 bytes)	DOUBLE PRECISION
	BINARY_DOUBLE	A 64-bit floating-point number (8 bytes)	DOUBLE PRECISION
	DATE	N/A	TIMESTAMP(0)

Category	Data type of Oracle	Value range	Data type of AnalyticDB for PostgreSQL
	TIMESTAMP [(fractional_seconds_pr ecision)]	N/A	TIMESTAMP
Date and time	TIMESTAMP [(fractional_seconds_pr ecision)] WITH TIME ZONE	N/A	TIMESTAMP WITH TIME ZONE
	TIMESTAMP [(fractional_seconds_pr ecision)] WITH LOCAL TIME ZONE	N/A	TIMESTAMP WITH TIME ZONE
	INTERVAL YEAR [(year_precision)] TO MONTH	N/A	VARCHAR(32)
	INTERVAL DAY [(day_precision)] TO SECOND [(fractional_seconds_pr ecision)]	N/A	VARCHAR(32)
	CHAR [(size [BYTE  CHAR])]	2,000 bytes	CHAR
	NCHAR[(size)]	2,000 bytes	VARCHAR
	VARCHAR2(size [BYTE  CHAR])	If MAX_STRING_SIZE is set to EXTENDED, the maximum size is 32,767 bytes. If MAX_STRING_SIZE is set to STANDARD, the maximum size is 4,000 bytes.	VARCHAR
	NVARCHAR2(size)	If MAX_STRING_SIZE is set to EXTENDED, the maximum size is 32,767 bytes. If MAX_STRING_SIZE is set to STANDARD, the maximum size is 4,000 bytes.	VARCHAR
	LONG	The maximum size is 2 GB (2^31 - 1).	TEXT

String

Category	Data type of Oracle	Value range	Data type of AnalyticDB for PostgreSQL
	RAW(size)	The maximum size is 32,767 bytes or 2,000 bytes.	BYTEA
	LONG RAW	The maximum size is 2 GB.	BYTEA
	CLOB	The maximum size is (4 GB - 1) × DB_BLOCK_SIZE.	ТЕХТ
	NCLOB	The maximum size is (4 GB - 1) × DB_BLOCK_SIZE.	ТЕХТ
	BLOB	The maximum size is (4 GB - 1) × DB_BLOCK_SIZE.	BYTEA
	BFILE	The maximum size is 4 GB.	Not supported
JSON	JSON	The maximum size is 32 MB.	JSON
ROWID	ROWID	64 characters	OID
Spatial	Customization required	Not supported	

**Note** If an Oracle data type is not supported by AnalyticDB for PostgreSQL, DTS converts the data type to BYTEA. If the conversion fails, DTS sets the field value to NULL.

## Synchronize data from a PostgreSQL database

The following table lists the data type mappings between PostgreSQL and AnalyticDB for PostgreSQL. The source PostgreSQL database can be a self-managed PostgreSQL database or an ApsaraDB RDS for PostgreSQL instance.

**Note** If the value range of the data to be synchronized from the source instance exceeds the range supported by DTS, the accuracy of the data written to the destination instance will decrease.

Category	Data type of PostgreSQL	Value range	Data type of AnalyticDB for PostgreSQL
	SMALLINT	-32767 to +32767	SMALLINT
	INT EGER	-2147483648 to +2147483647	INT EGER

Integer Category	Data type of PostgreSQL	Value range	Data type of AnalyticDB for PostgreSQL
	BIGINT	-9223372036854775808 to +9223372036854775807	BIGINT
	DECIMAL	Up to 131072 digits before the decimal point. Up to 16383 digits after the decimal point.	DECIMAL
Decimal	NUMERIC	Up to 131072 digits before the decimal point. Up to 16383 digits after the decimal point.	NUMERIC
	REAL	6 decimal digits of precision	REAL
	DOUBLE PRECISION	15 decimal digits of precision	DOUBLE PRECISION
Monetary	MONEY	-92233720368547758.08 to +92233720368547758.0 7	MONEY
	CHARACT ER VARYING(n)	N/A	CHARACT ER VARYING(n)
	CHARACT ER(n)	N/A	CHARACT ER(n)
String	TEXT	N/A	TEXT
String	CHAR	The default length is 1 byte.	CHAR
	NAME	The maximum length is 64 bytes.	NAME
	T SQUERY	A text query	TEXT
Text search	TSVECTOR	A document in a form optimized for text search	TEXT
Binary	BYTEA	1 or 4 bytes plus the actual binary string	BYTEA
	TIMESTAMP [ (p) ] [ WITHOUT TIME ZONE ]	Date and time without time zone. Storage size: 8 bytes.	TIMESTAMP [ (p) ] [ WITHOUT TIME ZONE ]

Category	Data type of PostgreSQL	Value range	Data type of AnalyticDB for PostgreSQL
	TIMESTAMP [ (p) ] WITH TIME ZONE	Date and time with time zone. Storage size: 8 bytes.	TIMESTAMP [ (p) ] WITH TIME ZONE
Date and time	DATE	A date. Storage size: 4 bytes.	DATE
	TIME [ (p) ] [ WITHOUT TIME ZONE ]	A time without time zone. Storage size: 8 bytes.	TIME [ (p) ] [ WITHOUT TIME ZONE ]
	TIME [ (p) ] WITH TIME ZONE	A time with time zone. Storage size: 12 bytes.	TIME [ (p) ] WITH TIME ZONE
	interval [ fields ] [ (p) ]	A time interval. Storage size: 16 bytes.	interval [ fields ] [ (p) ]
Boolean	BOOLEAN	1 byte	BOOLEAN
Enumerated	Customization required	N/A	VARCHAR(128)
	POINT	A point on a plane. Storage size: 16 bytes.	POINT
	LINE	An infinite line. Storage size: 32 bytes.	LINE
	LSEG	A finite line segment. Storage size: 32 bytes.	LSEG
Spatial	BOX	A rectangular box. Storage size: 32 bytes.	BOX
	PATH	A path. Storage size: 16 + 16n bytes.	PATH
	POLYGON	A polygon (similar to closed path). Storage size: 40 + 16n bytes.	POLYGON
	CIRCLE	A circle. Storage size: 24 bytes.	CIRCLE
	CIDR	IPv4 and IPv6 networks. Storage size: 7 or 19 bytes.	CIDR
	INET	IPv4 and IPv6 hosts and INET networks. Storage size: 7 or 19 bytes.	

Notwork addross

Category	Data type of PostgreSQL	Value range	Data type of AnalyticDB for PostgreSQL
	MACADDR	MAC addresses. Storage size: 6 bytes.	MACADDR
	MACADDR8	MAC addresses in EUI-64 format. Storage size: 8 bytes.	MACADDR8
	Bit (n)	N/A	Bit (n)
ысыны	BIT VARYING (n)	N/A	BIT VARYING (n)
UUID	UUID	N/A	VARCHAR(64)
XML	XML	N/A	XML
JSON	JSON	N/A	JSON
	JSONB	N/A	JSONB

## Synchronize data from an SQL Server database

The following table lists the data type mappings between SQL Server and AnalyticDB databases. The source instance can be a self-managed SQL Server database or an ApsaraDB RDS for SQL Server instance. The destination instance can be an AnalyticDB for MySQL cluster or an AnalyticDB for PostgreSQL instance.

**Note** If the value range of the data to be synchronized from the source instance exceeds the range supported by DTS, the accuracy of the data written to the destination instance will decrease.

Category	Data type of SQL Server	Value range	Data type of AnalyticDB for MySQL	Data type of AnalyticDB for PostgreSQL
	BIT	An INTEGER data type that can take a value of 1, 0, or NULL	BOOLEAN	BIT(1)
	TINYINT	0 to 255	TINYINT	SMALLINT
	SMALLINT	-32,768 (-2^15) to 32,767 (2^15 - 1)	SMALLINT	SMALLINT
Integer	INT	-2,147,483,648 (- 2^31) to 2,147,483,647 (2^31 - 1)	INT EGER	INT EGER

Category	Data type of SQL Server	Value range	Data type of AnalyticDB for MySQL	Data type of AnalyticDB for PostgreSQL
	BIGINT	- 9,223,372,036,854, 775,808 (-2^63) to 9,223,372,036,854, 775,807 (2^63 - 1)	BIGINT	BIGINT
	NUMERIC[ (p[ ,s] )]	-10^38 + 1 to 10^38 - 1 (1 <= p <= 38)	DECIMAL	DECIMAL
	DECIMAL[ (p[ ,s] )]	-10^38 + 1 to 10^38 - 1 (1 <= p <= 38)	DECIMAL	DECIMAL
Decimal	FLOAT	-1.79E + 308 to - 2.23E - 308, 0, and 2.23E - 308 to 1.79E + 308	DOUBLE	DOUBLE PRECISION
	REAL	-3.40E + 38 to - 1.18E - 38, 0, and 1.18E - 38 to 3.40E + 38	FLOAT	REAL
Monetary	MONEY	- 922,337,203,685,4 77.5808 to 922,337,203,685,4 77.5807	DECIMAL(19, 4)	DECIMAL(19, 4)
	SMALLMONEY	-214,748.3648 to 214,748.3647	DECIMAL(10, 4)	DECIMAL(10, 4)
	DATE	0001-01-01 to 9999-12-31	DATE	DATE
	DAT ET IME	Date range: January 1, 1753 to December 31, 9999 Time range: 00:00:00 to 23:59:59.997	DATETIME	TIMESTAMP(3) WITHOUT TIME ZONE
	DAT ET IME2[ (fractional seconds precision) ]	Date range: January 1,1 CE to December 31, 9999 CE Time range: 00:00:00 to 23:59:59.99999999	DAT ET IME	TIMESTAMP(7) WITHOUT TIME ZONE

Category	Data type of SQL Server	Value range	Data type of AnalyticDB for MySQL	Data type of AnalyticDB for PostgreSQL
	DAT ET IMEOFFSET [ (fractional seconds precision) ]	Date range: January 1, 1 CE to December 31, 9999 CE Time range: 00:00:00 to 23:59:59.99999999 Time zone offset range: -14:00 to +14:00	T IMEST AMP	T IMEST AMP(7) WIT H T IME ZONE
	SMALLDAT ET IME	The time is based on a 24-hour day, with seconds always zero (:00) and without fractional seconds.	DAT ET IME	TIMESTAMP WITHOUT TIME ZONE
	TIME [ (fractional second scale) ]	00:00:00.0000000 to 23:59:59.99999999	TIME	TIME(7) WITH TIME ZONE
	BINARY [ ( n ) ]	Valid values of n: 1 to 8,000.	VARBINARY	BYTEA
	VARBINARY [ ( n   max) ]	Valid values of n: 1 to 8,000. max indicates that the maximum storage size is 2^31 - 1 bytes.	VARBINARY	BYTEA
	CHAR [ ( n ) ]	Valid values of n: 1 to 8,000. The storage size is n bytes.	VARCHAR	CHARACT ER
	VARCHAR [ ( n   max ) ]	Valid values of n: 1 to 8,000. max indicates that the maximum storage size is 2^31 - 1 bytes (2 GB).	VARCHAR	CHARACT ER

Category	Data type of SQL Server	Value range	Data type of AnalyticDB for MySQL	Data type of AnalyticDB for PostgreSQL
String	NCHAR [ ( n ) ]	n defines the string size in byte- pairs. Valid values of n: 1 to 4,000. The storage size is two times n bytes.	VARCHAR	CHARACT ER VARYING
	NVARCHAR [ ( n   max ) ]	n defines the string size in byte- pairs. Valid values of n: 1 to 1 to 4,000. max indicates that the maximum storage size is 2^30 - 1 characters (2 GB).	VARCHAR	CHARACT ER VARYING
	NTEXT	Variable-length Unicode data with a maximum string length of 1,073,741,823 (2^30 - 1) bytes.	VARCHAR	TEXT
	TEXT	The maximum string length is 2,147,483,647 (2^31 - 1) bytes.	VARCHAR	TEXT
	IMAGE	Variable-length binary data from 0 to 2,147,483,647 (2^31 - 1) bytes.	VARBINARY	BYTEA
Spatial	GEOGRAPHY	N/A	VARCHAR	Not supported
geometry)	GEOMET RY	N/A	VARCHAR	Not supported
XML	XML ( [ CONTENT   DOCUMENT ] xml_schema_colle ction )	N/A	VARCHAR	XML
	UNIQUEIDENT IFIER	N/A	VARCHAR	CHARACT ER(36)
	SQL_VARIANT	N/A	Not supported	Not supported
	HIERARCHYID	N/A	Not supported	Not supported
Others				

Category	Data type of SQL Server	Value range	Data type of AnalyticDB for MySQL	Data type of AnalyticDB for PostgreSQL
	SYSNAME	N/A	VARCHAR	CHARACTER VARYING(128)

# Synchronize data from a Db2 for LUW database

The following table lists the data type mappings between a Db2 for LUW database and a MySQL database.

**Note** If the value range of the data to be synchronized from the source instance exceeds the range supported by DTS, the accuracy of the data written to the destination instance will decrease.

Category	Data type of Db2 for LUW	Value range	Data type of MySQL
	SMALLINT	-32,768~+32,767	SMALLINT
Integer	INTEGER	- 2,147,483,648~+2,147,4 83,647	INT
	BIGINT	- 9,223,372,036,854,775,8 08~ +9,223,372,036,854,775, 807	BIGINT
Decimal	DECIMAL(precision- integer, scale-integer)	p<=38	DECIMAL
	FLOAT (integer)	The value range is 1 to 53. If the integer is between 1 and 24 inclusive, the format is single precision floating-point. If the integer is between 25 and 53 inclusive, the format is double precision floating-point.	FLOAT
	DECFLOAT (precision- integer)	N/A	DECIMAL(65,10)
	DATE	0001-01-01~9999-12-31	DATE
	TIME	00:00:00~24:00:00	TIME

Date and time Category	Data type of Db2 for LUW	Value range	Data type of MySQL
	TIMESTAMP(integer)	0001-01-01- 00.00.00.00000000000 ~9999-12-31- 24.00.00.000000000000; 0<=p<= 12	DAT ET IME
	CHARACT ER(int eger)	254	CHAR   VARCHAR
String	VARCHAR(integer)	32,672	VARCHAR
	CHARACTER(integer) FOR BIT DATA	254	BLOB
	CLOB	2,147,483,647	LONGT EXT
	GRAPHIC(integer)	127	CHAR(length*4)
	VARGRAPHIC(integer)	16,336	CHAR(length*4)
	DBCLOB(integer)	1,073,741,823	VARCHAR   LONGT EXT
	BLOB	2,147,483,647	LONGBLOB
Others	XML	2,147,483,647	VARCHAR   LONGT EXT

## Synchronize data from a Db2 for i database

The following table lists the data type mappings between a Db2 for i database and a MySQL database.

**Note** If the value range of the data to be synchronized from the source instance exceeds the range supported by DTS, the accuracy of the data written to the destination instance will decrease.

Category	Data type of Db2 for i	Value range	Data type of MySQL
Integer	SMALLINT	-32,768~+32,767	SMALLINT
	INTEGER	- 2,147,483,648~+2,147,4 83,647	INT
	BIGINT	- 9,223,372,036,854,775,8 08~ +9,223,372,036,854,775, 807	BIGINT
	DECIMAL(precision- integer, scale-integer)	p<=63	DECIMAL

Category	Data type of Db2 for i	Value range	Data type of MySQL
Decimal	NUMERIC	N/A	DECIMAL
	FLOAT (integer)	N/A	FLOAT
	DECFLOAT (precision- integer)	N/A	DECIMAL(65,10)
	DATE	0001-01-0~9999-12-31	DATE
	TIME	00:00:00~24:00:00	TIME
Date and time	TIMESTAMP(integer)	0001-01-01- 00.00.00.00000000000000000 to 9999-12-31- 24.00.00.0000000000000000 (0 <= p <= 12)	DATETIME
	CHAR(integer)	32,765	CHAR   VARCHAR
	VARCHAR(integer)	32,739	VARCHAR
	CHAR(integer) FOR BIT DAT A	N/A	BLOB
	CLOB	2,147,483,647	LONGTEXT
String	GRAPHIC(integer)	16,382	CHAR
	VARGRAPHIC(integer)	16,369	VARCHAR
	DBCLOB(integer)	1,073,741,823	LONGTEXT
	BINARY	32,765	BINARY
	VARBIN	32,739	VARBINARY
	BLOB	2,147,483,647	LONGBLOB
	DATALINK	N/A	VARCHAR   LONGT EXT
Others	ROWID	40	VARCHAR   LONGT EXT
	XML	2,147,483,647	VARCHAR   LONGT EXT

# 3.Precautions and limits 3.1. Precautions and limits for synchronizing data from a MySQL database

This topic describes the precautions and limits when you synchronize data from MySQL databases, such as self-managed MySQL databases and ApsaraDB RDS for MySQL instances. To ensure that your data synchronization task runs as expected, you must read the precautions and limits before you configure the task.

# Scenarios of synchronizing data from a MySQL database

Take note of precautions and limits in the following data synchronization scenarios:

#### ? Note

By default, Data Transmission Service (DTS) disables FOREIGN KEY constraints for the destination database in a data synchronization task. Therefore, the cascade and delete operations of the source database are not synchronized to the following destination databases:

- MySQL (ApsaraDB RDS for MySQL instance and self-managed MySQL database)
- PolarDB for MySQL cluster
- AnalyticDB for MySQL cluster
- AnalyticDB for PostgreSQL instance
- Elasticsearch cluster
- ApsaraDB for ClickHouse cluster
- Tablestore instance
- PolarDB-X 2.0 instance
- Synchronize data between MySQL databases
- Synchronize data from a MySQL database to a PolarDB for MySQL cluster
- Synchronize data from a MySQL database to an AnalyticDB for MySQL cluster
- Synchronize data from a MySQL database to an AnalyticDB for PostgreSQL instance
- Synchronize data from a MySQL database to a DataHub project
- Synchronize data from a MySQL database to an Elasticsearch cluster
- Synchronize data from a MySQL database to a MaxCompute project
- Synchronize data from a MySQL database to an ApsaraDB for ClickHouse cluster
- Synchronize data from a MySQL database to a Tablestore instance
- Synchronize dat a from a MySQL dat abase to a Message Queue for Apache Kaf ka instance or a selfmanaged Kaf ka clust er
- Synchronize data from a MySQL database to a PolarDB-X instance

#### Synchronize data between MySQL databases

The following table describes the precautions and limits when you synchronize data between MySQL
databases, such as self-managed MySQL databases and ApsaraDB RDS for MySQL instances.

• One-way dat a synchronization between MySQL dat abases

Category	Description
Limits on the source database	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature is enabled. For more information about how to enable binary logging, see Modify the parameters of an ApsaraDB RDS for MySQL instance. In addition, the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	<ul> <li>Notice</li> <li>If the source database is a self-managed MySQL database, you must enable binary logging and set binlog_format to row and binlog_row_image to full.</li> <li>If the self-managed MySQL database is deployed in a dual-primary cluster, you must set log_slave_updates to ON. This ensures that DTS can obtain all binary logs. For more information, see Create an account for a user-created MySQL database and configure binary logging.</li> </ul>
	For an incremental data synchronization task, the binary logs of the source database are retained for at least 24 hours. For a full data and incremental data synchronization task, the binary logs of the source database are retained for at least seven days. After full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance. For more information about binary log files and log backup files of an ApsaraDB RDS for MySQL instance.

Category	Description
Other limits	<ul> <li>To ensure compatibility, the version of the destination database must be the same as or later than that of the source database. If the version of the destination database is earlier than that of the source database, database compatibility issues may occur.</li> </ul>
	<ul> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads on the database servers.</li> </ul>
	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is complete, the size of the used tablespace of the destination database is larger than that of the source database.
	<ul> <li>If you select one or more tables instead of an entire database as the objects to synchronize, do not use gh-ost or pt-online-schema-change to perform DDL operations on the tables during data synchronization. Otherwise, data may fail to be synchronized. If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables during data synchronization. For more information, see Change schemas without locking tables.</li> </ul>
	<ul> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.</li> </ul>
	If the source database is a self-managed MySQL database, take note of the following items:
Precaution s	<ul> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> </ul>
	• DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the synchronization latency is too high, you can perform a DML operation on the source database to update the latency.
	<b>Note</b> If you select an entire database as the object to synchronize, you can create a heartbeat table. The heartbeat table is updated or receives data every second.

### • Two-way data synchronization between MySQL databases

Category

Category	Description
	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature is enabled. For more information about how to enable binary logging, see Modify the parameters of an ApsaraDB RDS for MySQL instance. In addition, the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
Limits on the source and destination databases	<ul> <li>Notice</li> <li>If the source database is a self-managed MySQL database, you must enable binary logging and set binlog_format to row and binlog_row_image to full.</li> <li>If the self-managed MySQL database is deployed in a dual-primary cluster, you must set log_slave_updates to ON. This ensures that DTS can obtain all binary logs. For more information, see Create an account for a user-created MySQL database and configure binary logging.</li> </ul>
	For an incremental data synchronization task, the binary logs of the source database are retained for at least 24 hours. For a full data and incremental data synchronization task, the binary logs of the source database are retained for at least seven days. After full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance. For more information about binary log files and log backup files of an ApsaraDB RDS for MySQL instance, see Introduction to binary log files and log backup files of an ApsaraDB RDS for MySQL instance.

Category	Description
	<ul> <li>To ensure compatibility, the version of the destination database must be the same as or later than that of the source database. If the version of the destination database is earlier than that of the source database, database compatibility issues may occur.</li> </ul>
	<ul> <li>When DTS runs a two-way data synchronization task, DTS creates a database named dts in the destination database to prevent circular synchronization. When the task is running, do not modify the dts database.</li> </ul>
	<ul> <li>If the source or destination database is located in a region outside the Chinese mainland, two-way data synchronization is supported only between databases located within the same region. For example, two-way data synchronization is supported between databases within the Japan (Tokyo) region. Two-way data synchronization between a database in the Japan (Tokyo) region and another database in the Germany (Frankfurt) region is not supported.</li> </ul>
	<ul> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads on the database servers.</li> </ul>
Other limits	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is complete, the size of the used tablespace of the destination database is larger than that of the source database.
	<ul> <li>If you select one or more tables instead of an entire database as the objects to synchronize, do not use gh-ost or pt-online-schema-change to perform DDL operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables during data</li> </ul>
	<ul> <li>synchronization. For more information, see Change schemas without locking tables.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.</li> </ul>
	<ul> <li>By default, DTS disables FOREIGN KEY constraints for the destination database in a data synchronization task. Therefore, the cascade and delete operations of the source database are not synchronized to the destination database.</li> </ul>
	• For a table that is synchronized in both the forward and reverse synchronization and both the full data and incremental data of the table are synchronized in the forward synchronization, DTS synchronizes only the incremental data of the table in the reverse synchronization.

Category	Description
Special cases	<ul> <li>If the source database is a self-managed MySQL database, take note of the following items:</li> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the synchronization latency is too high, you can perform a DML operation on the source database to update the latency.</li> </ul>
	<b>Note</b> If you select an entire database as the object to synchronize, you can create a heartbeat table. The heartbeat table is updated or receives data every second.

# Synchronize data from a MySQL database to a PolarDB for MySQL cluster

• One-way data synchronization from a MySQL database to a PolarDB for MySQL cluster

Category

Category	Description
	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging, see Modify the parameters of an ApsaraDB RDS for MySQL instance. In addition, the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
Limits on the source database	<ul> <li>Notice</li> <li>If the source database is a self-managed MySQL database, you must enable binary logging and set binlog_format to row and binlog_row_image to full.</li> <li>If the self-managed MySQL database is deployed in a dual-primary cluster, you must set log_slave_updates to ON. This ensures that DTS can obtain all binary logs. For more information, see Create an account for a user-created MySQL database and configure binary logging.</li> </ul>
	For an incremental data synchronization task, the binary logs of the source database are retained for at least 24 hours. For a full data and incremental data synchronization task, the binary logs of the source database are retained for at least seven days. After full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance. For more information about binary log files and log backup files of an ApsaraDB RDS for MySQL instance, see Introduction to binary log files and log backup files of an ApsaraDB RDS for MySQL instance.

Category	Description
Other limits	<ul> <li>To ensure compatibility, the version of the destination database must be the same as or later than that of the source database. If the version of the destination database is earlier than that of the source database, database compatibility issues may occur.</li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads on the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is complete, the size of the used tablespace of the destination database is larger than that of the source database.</li> <li>If you select one or more tables instead of an entire database as the objects to synchronized.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables during data synchronization. For more information, see Change schemas without locking tables.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination database.</li> </ul>
Special cases	<ul> <li>If the source database is a self-managed MySQL database, take note of the following items:</li> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the synchronization latency is too high, you can perform a DML operation on the source database to update the latency.</li> <li>Note If you select an entire database as the object to synchronize, you can create a heartbeat table. The heartbeat table is updated or receives data every second.</li> </ul>

• Two-way data synchronization from a MySQL database to a PolarDB for MySQL cluster

Category	Description
	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> </ul>
	<ul> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> </ul>
	• The following requirements for binary logs must be met:
	<ul> <li>For the MySQL database: The binary logging feature is enabled. For more information about how to enable binary logging, see Modify the parameters of an ApsaraDB RDS for MySQL instance. In addition, the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	5) Notice
	<ul> <li>If the source database is a self-managed MySQL database, you must enable binary logging and set binlog_format to row and binlog_row_image to full.</li> </ul>
	If the self-managed MySQL database is deployed in a dual-primary cluster, you must set log_slave_updates to ON. This ensures that DTS can obtain all binary logs. For more information, see Create an account for a user-created MySQL database and configure binary logging.
	For the PolarDB for MySQL cluster: The binary logging feature must be enabled. The loose_polar_log_bin parameter must be set to on. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.
	For an incremental data synchronization task, the binary logs of the source database are retained for at least 24 hours. For a full data and incremental data synchronization task, the binary logs of the source database are retained for at least seven days. After full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the
Limits on the source and destination databases	binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance. For more information about binary log files and log backup files of an ApsaraDB RDS for MySQL instance, see Introduction to binary log files and log backup files of an ApsaraDB RDS for MySQL instance.

Category	Description

Category	Description
Other limits	<ul> <li>When DTS runs a two-way data synchronization task, DTS creates a database named dts in the destination database to prevent circular synchronization. When the task is running, do not modify the dts database.</li> <li>If the source or destination database is located in a region outside the Chinese mainland, two-way data synchronization is supported only between databases located within the same region. For example, two-way data synchronization is supported between databases within the Japan (Tokyo) region. Two-way data synchronization between a database in the Japan (Tokyo) region and another database in the Germany (Frankfurt) region is not supported.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.</li> <li>By default, DTS disables FOREIGN KEY constraints for the destination database in a data synchronization task. Therefore, the cascade and delete operations of the source database are not synchronized to the destination database.</li> <li>For a table that is synchronized in both the forward and reverse synchronization and both the full data and incremental data of the table are synchronization.</li> </ul>
Special cases	<ul> <li>If the source database is a self-managed MySQL database, take note of the following items:</li> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the synchronization latency is too high, you can perform a DML operation on the source database to update the latency.</li> <li>Note If you select an entire database as the object to synchronize, you can create a heartbeat table. The heartbeat table is updated or receives data every second.</li> </ul>

# Synchronize data from a MySQL database to an AnalyticDB for MySQL cluster

Category	Description
Limits on the source database	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature is enabled. For more information about how to enable binary logging, see Modify the parameters of an ApsaraDB RDS for MySQL instance. In addition, the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	<ul> <li>Notice</li> <li>If the source database is a self-managed MySQL database, you must enable binary logging and set binlog_format to row and binlog_row_image to full.</li> <li>If the self-managed MySQL database is deployed in a dual-primary cluster, you must set log_slave_updates to ON. This ensures that DTS can obtain all binary logs. For more information, see Create an account for a user-created MySQL database and configure binary logging.</li> </ul>
	<ul> <li>For an incremental data synchronization task, the binary logs of the source database are retained for at least 24 hours. For a full data and incremental data synchronization task, the binary logs of the source database are retained for at least seven days. After full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance. For more information about binary log files and log backup files of an ApsaraDB RDS for MySQL instance, see Introduction to binary log files and log backup files of an ApsaraDB RDS for MySQL instance.</li> </ul>
	• During synchronization, do not perform DDL operations to modify the primary key and add comments (such as ALTER TABLE table_name COMMENT=' comments on the table'; ), Otherwise, the DDL execution will fail during data synchronization.

Description
<ul> <li>Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to be synchronized.</li> <li>Due to the limits of , if the disk space usage of the nodes in an cluster reaches 80%, the task is delayed and error messages are returned. We recommend that you estimate the required disk space based on the objects to synchronize. You must ensure that the destination cluster has sufficient storage space.</li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads on the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is complete, the size of the used tablespace of the destination database is larger than that of the source database.</li> <li>If you select one or more tables instead of an entire database as the objects to synchronize, do not use gh-ost or pt-online-schema-change to perform DDL operations on the tables during data synchronization. Otherwise, data may fail to be synchronized. If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables during data synchronization, see Change schemas without locking tables.</li> <li>If you use only DTS to write data to the destination database in the cluster, you can use DMS to perform online DDL operations on source tables during data synchronization. For more information, see Change schemas without locking tables.</li> <li>If you use only DTS to write data to the destination database.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database.</li> <li>During data synchronizati</li></ul>
<ul> <li>If the source database is a self-managed MySQL database, take note of the following items:</li> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the synchronization latency is too high, you can perform a DML operation on the source database to update the latency.</li> <li>Note If you select an entire database as the object to synchronize, you can create a heartbeat table. The heartbeat table is updated or receives data every second.</li> </ul>

# Synchronize data from a MySQL database to an AnalyticDB for PostgreSQL instance

Category	Description
Limits on the source database	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature must be enabled. For more information about how to enable binary logging, see Modify the parameters of an ApsaraDB RDS for MySQL instance. In addition, the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	<ul> <li>Notice</li> <li>If the source database is a self-managed MySQL database, you must enable binary logging and set binlog_format to row and binlog_row_image to full.</li> <li>If the self-managed MySQL database is deployed in a dual-primary cluster, you must set log_slave_updates to ON. This ensures that DTS can obtain all binary logs. For more information, see Create an account for a user-created MySQL database and configure binary logging.</li> </ul>
	<ul> <li>For an incremental data synchronization task, the binary logs of the source database are retained for at least 24 hours. For a full data and incremental data synchronization task, the binary logs of the source database are retained for at least seven days. After full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the SLA of DTS does not guarantee service reliability and performance. For more information about binary log files and log backup files of an ApsaraDB RDS for MySQL instance.</li> <li>During data synchronization, do not perform DDL operations to modify the primary key or</li> </ul>
	<ul> <li>During data synchronization, do not perform DDL operations to modify the primary key or add comments such as ALTER TABLE table_name COMMENT='Table comments'; because the operations cannot take effect.</li> </ul>

Category	Description
Other limits	<ul> <li>Requirements for the objects to synchronize: <ul> <li>Only tables can be selected as the objects to synchronize.</li> <li>DTS does not synchronize the following data types: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID_SNAPSHOT.</li> <li>Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to be synchronized.</li> </ul> </li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads on the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is complete, the size of the used tablespace of the destination database is larger than that of the source database.</li> <li>If you select one or more tables instead of an entire database as the objects to synchronized. If you use only DT S to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables during data synchronization. Otherwise, data may fail to be synchronized. If you use only DT S to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables during data synchronization, we recommend that you use only DT S to write data to the destination database, source and destination database.</li> <li>During data synchronization, we recommend that you use only DT S to write data to the destination database, data lot the destination database, data lots may occur in the destination database when you use DMS to perform online DDL operations.</li> </ul>
Special cases	<ul> <li>If the source database is a self-managed MySQL database, take note of the following items:</li> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the synchronization latency is too high, you can perform a DML operation on the source database to update the latency.</li> <li>Note If you select an entire database as the object to synchronize, you can create a heartbeat table. The heartbeat table is updated or receives data every second.</li> </ul>

## Synchronize data from a MySQL database to a DataHub project

Category	Description
Limits on the source database	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature is enabled. For more information about how to enable binary logging, see Modify the parameters of an ApsaraDB RDS for MySQL instance. In addition, the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	<ul> <li>Notice</li> <li>If the source database is a self-managed MySQL database, you must enable binary logging and set binlog_format to row and binlog_row_image to full.</li> <li>If the self-managed MySQL database is deployed in a dual-primary cluster, you must set log_slave_updates to ON. This ensures that DTS can obtain all binary logs. For more information, see Create an account for a user-created MySQL database and configure binary logging.</li> </ul>
	<ul> <li>For an incremental data synchronization task, the binary logs of the source database are retained for at least 24 hours. For a schema and incremental data synchronization task, the binary logs of the source database are retained for at least seven days. After the schema is synchronized, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the SLA of DTS does not ensure service reliability and performance. For more information about binary log files and log backup files of an ApsaraDB RDS for MySQL instance, see Introduction to binary log files and log backup files of an ApsaraDB RDS for MySQL instance.</li> </ul>
Ot her limit s	<ul> <li>Initial full data synchronization is not supported. DTS does not synchronize the historical data of required objects from the source ApsaraDB RDS for MySQL instance to the destination DataHub instance.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.</li> </ul>

Category	Description
Special cases	<ul> <li>If the source database is a self-managed MySQL database, take note of the following items:</li> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the synchronization latency is too high, you can perform a DML operation on the source database to update the latency.</li> </ul>
	<b>Note</b> If you select an entire database as the object to synchronize, you can create a heartbeat table. The heartbeat table is updated or receives data every second.

## Synchronize data from a MySQL database to an Elasticsearch cluster

Category	
Calegory	

Category	Description
	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature is enabled. For more information about how to enable binary logging, see Modify the parameters of an ApsaraDB RDS for MySQL instance. In addition, the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
Limits on the source database	<ul> <li>Notice         <ul> <li>If the source database is a self-managed MySQL database, you must enable binary logging and set binlog_format to row and binlog_row_image to full.</li> <li>If the self-managed MySQL database is deployed in a dual-primary cluster, you must set log_slave_updates to ON. This ensures that DTS can obtain all binary logs. For more information, see Create an account for a user-created MySQL database and configure binary loggs of the source database are retained for at least 24 hours. For a full data and incremental data synchronization task, the binary logs of the source database are retained for at least 24 hours. For a full data and incremental data synchronization task, the binary logs of the source database are retained for at least seven days. After full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance. For more information about binary log files and log backup files of an ApsaraDB RDS for MySQL instance.</li> </ul> </li> </ul>

Category	Description
Other limits	<ul> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads on the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is complete, the size of the used tablespace of the destination database is larger than that of the source database.</li> <li>If you select one or more tables instead of an entire database as the objects to synchronize, do not use gh-ost or pt-online-schema-change to perform DDL operations on the tables during data synchronization. Otherwise, data may fail to be synchronized. If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables during data synchronization, see Change schemas without locking tables.</li> <li>To add columns to a table that you want to synchronize, perform DDL operations in the source MySQL database, and then pause and start the data synchronization task.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination database. The source inform and the pause and start the data to the destination database.</li> </ul>
Special cases	<ul> <li>If the source database is a self-managed MySQL database, take note of the following items:</li> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the synchronization latency is too high, you can perform a DML operation on the source database to update the latency.</li> <li>Note If you select an entire database as the object to synchronize, you can create a heartbeat table. The heartbeat table is updated or receives data every second.</li> </ul>

## Synchronize data from a MySQL database to a MaxCompute project

Category	Description
Limits on the source database	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature is enabled. For more information about how to enable binary logging, see Modify the parameters of an ApsaraDB RDS for MySQL instance. In addition, the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	<ul> <li>Notice</li> <li>If the source database is a self-managed MySQL database, you must enable binary logging and set binlog_format to row and binlog_row_image to full.</li> <li>If the self-managed MySQL database is deployed in a dual-primary cluster, you must set log_slave_updates to ON. This ensures that DTS can obtain all binary logs. For more information, see Create an account for a user-created MySQL database and configure binary logging.</li> </ul>
	<ul> <li>For an incremental data synchronization task, the binary logs of the source database are retained for at least 24 hours. For a full data and incremental data synchronization task, the binary logs of the source database are retained for at least seven days. After full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance. For more information about binary log files and log backup files of an ApsaraDB RDS for MySQL instance.</li> </ul>

Category	Description
Other limits	<ul> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads on the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is complete, the size of the used tablespace of the destination database is larger than that of the source database.</li> <li>If you select one or more tables instead of an entire database as the objects to synchronized, do not use gh-ost or pt-online-schema-change to perform DDL operations on the tables during data synchronization. Otherwise, data may fail to be synchronized. If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables during data synchronize no, see Change schemas without locking tables.</li> <li>MaxCompute does not support PRIMARY KEY constraints. If network errors occur, DTS may synchronize duplicate data records to the MaxCompute project.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. If you use tools other than DTS to write data to the destination database.</li> </ul>
	database, data loss may occur in the destination database when you use DMS to perform online DDL operations.
Special cases	<ul> <li>If the source database is a self-managed MySQL database, take note of the following items:</li> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the synchronization latency is too high, you can perform a DML operation on the source database to update the latency.</li> </ul>
	<b>Note</b> If you select an entire database as the object to synchronize, you can create a heartbeat table. The heartbeat table is updated or receives data every second.

# Synchronize data from a MySQL database to an ApsaraDB for ClickHouse cluster

Category

Category	Description
Limits on the source database	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature is enabled. For more information about how to enable binary logging, see Modify the parameters of an ApsaraDB RDS for MySQL instance. In addition, the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	<ul> <li>Notice</li> <li>If the source database is a self-managed MySQL database, you must enable binary logging and set binlog_format to row and binlog_row_image to full.</li> <li>If the self-managed MySQL database is deployed in a dual-primary cluster, you must set log_slave_updates to ON. This ensures that DTS can obtain all binary logs. For more information, see Create an account for a user-created MySQL database and configure binary logging.</li> </ul>
	<ul> <li>For an incremental data synchronization task, the binary logs of the source database are retained for at least 24 hours. For a full data and incremental data synchronization task, the binary logs of the source database are retained for at least seven days. After full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance. For more information about binary log files and log backup files of an ApsaraDB RDS for MySQL instance.</li> </ul>

Category	Description
	<ul> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads on the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the</li> </ul>
	tables of the destination database. After full data synchronization is complete, the size of the used tablespace of the destination database is larger than that of the source database.
Precaution s	<ul> <li>If you select one or more tables instead of an entire database as the objects to synchronize, do not use gh-ost or pt-online-schema-change to perform DDL operations on the tables during data synchronization. Otherwise, data may fail to be synchronized. If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables during data synchronization, see Change schemas without locking tables.</li> </ul>
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.
	• The names of the databases, tables, and columns to synchronize must comply with the naming conventions of ApsaraDB for ClickHouse. For more information, see the "Object naming conventions" section of the Limits topic.
	If the source database is a self-managed MySQL database, take note of the following items:
	<ul> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> </ul>
Special cases	• DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the synchronization latency is too high, you can perform a DML operation on the source database to update the latency.
	<b>Note</b> If you select an entire database as the object to synchronize, you can create a heartbeat table. The heartbeat table is updated or receives data every second.

## Synchronize data from a MySQL database to a Tablestore instance

Category

Category	Description
Limits on the source database	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature is enabled. For more information about how to enable binary logging, see Modify the parameters of an ApsaraDB RDS for MySQL instance. In addition, the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	<ul> <li>Notice</li> <li>If the source database is a self-managed MySQL database, you must enable binary logging and set binlog_format to row and binlog_row_image to full.</li> <li>If the self-managed MySQL database is deployed in a dual-primary cluster, you must set log_slave_updates to ON. This ensures that DTS can obtain all binary logs. For more information, see Create an account for a user-created MySQL database and configure binary logging.</li> </ul>
	<ul> <li>For an incremental data synchronization task, the binary logs of the source database are retained for at least 24 hours. For a full data and incremental data synchronization task, the binary logs of the source database are retained for at least seven days. After full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance. For more information about binary log files and log backup files of an ApsaraDB RDS for MySQL instance.</li> </ul>

Category	Description
Other limits	<ul> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads on the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is complete, the size of the used tablespace of the destination database is larger than that of the source database.</li> <li>If you select one or more tables instead of an entire database as the objects to synchronize, do not use gh-ost or pt-online-schema-change to perform DDL operations on the tables during data synchronization. Otherwise, data may fail to be synchronized. If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables during data synchronization, we recommend that you use only DTS to write data to the destination database.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. If you use tools other than DTS to write data to the destination database. If you use tools other than DTS to write data to the destination database.</li> <li>The names of the tables and columns to synchronize must comply with the naming conventions of Tablestore.</li> <li>The name of a table or an index can contain letters, digits, and underscores (_). The name must start with a letter or underscore (_).</li> </ul>
Special cases	<ul> <li>If the source database is a self-managed MySQL database, take note of the following items:</li> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the synchronization latency is too high, you can perform a DML operation on the source database to update the latency.</li> <li>Note If you select an entire database as the object to synchronize, you can create a heartbeat table. The heartbeat table is updated or receives data every second.</li> </ul>

## Synchronize data from a MySQL database to a Message Queue for Apache Kafka instance or a self-managed Kafka cluster

Category

Category	Description
Limits on the source database	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature is enabled. For more information about how to enable binary logging, see Modify the parameters of an ApsaraDB RDS for MySQL instance. In addition, the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	<ul> <li>Notice</li> <li>If the source database is a self-managed MySQL database, you must enable binary logging and set binlog_format to row and binlog_row_image to full.</li> <li>If the self-managed MySQL database is deployed in a dual-primary cluster, you must set log_slave_updates to ON. This ensures that DTS can obtain all binary logs. For more information, see Create an account for a user-created MySQL database and configure binary logging.</li> </ul>
	<ul> <li>For an incremental data synchronization task, the binary logs of the source database are retained for at least 24 hours. For a full data and incremental data synchronization task, the binary logs of the source database are retained for at least seven days. After full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance. For more information about binary log files and log backup files of an ApsaraDB RDS for MySQL instance.</li> </ul>

Category	Description
Precaution s	• Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads on the database servers.
	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is complete, the size of the used tablespace of the destination database is larger than that of the source database.
	<ul> <li>If you select one or more tables instead of an entire database as the objects to synchronize, do not use gh-ost or pt-online-schema-change to perform DDL operations on the tables during data synchronization. Otherwise, data may fail to be synchronized.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables during data synchronization, see Change schemas without locking tables.</li> </ul>
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.
	If the source database is a self-managed MySQL database, take note of the following items:
Special cases	<ul> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> </ul>
	• DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the synchronization latency is too high, you can perform a DML operation on the source database to update the latency.
	<b>Note</b> If you select an entire database as the object to synchronize, you can create a heartbeat table. The heartbeat table is updated or receives data every second.

# Synchronize data from a MySQL database to a PolarDB-X 2.0 instance

Category	Description
Limits on the source database	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature is enabled. For more information about how to enable binary logging, see Modify the parameters of an ApsaraDB RDS for MySQL instance. In addition, the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	<ul> <li>Notice</li> <li>If the source database is a self-managed MySQL database, you must enable binary logging and set binlog_format to row and binlog_row_image to full.</li> <li>If the self-managed MySQL database is deployed in a dual-primary cluster, you must set log_slave_updates to ON. This ensures that DTS can obtain all binary logs. For more information, see Create an account for a user-created MySQL database and configure binary logging.</li> </ul>
	<ul> <li>For an incremental data synchronization task, the binary logs of the source database are retained for at least 24 hours. For a full data and incremental data synchronization task, the binary logs of the source database are retained for at least seven days. After full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance. For more information about binary log files and log backup files of an ApsaraDB RDS for MySQL instance.</li> </ul>

Category	Description
Precaution s	<ul> <li>Schema synchronization is not supported in this scenario. Before you configure a data synchronization task, you must create databases and tables in the destination instance.</li> <li>Requirements for the objects to synchronize: <ul> <li>DTS does not synchronize the following data types: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID_SNAPSHOT.</li> <li>Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to be synchronized.</li> </ul> </li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads on the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is complete, the size of the used tablespace of the destination database is larger than that of the source database.</li> <li>If you select one or more tables instead of an entire database as the objects to synchronized. If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DL operations on source tables during data synchronization, see Change schemas without locking tables.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. If you use only DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform on and the database.</li> </ul>
Special cases	<ul> <li>If the source database is a self-managed MySQL database, take note of the following items:</li> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the synchronization latency is too high, you can perform a DML operation on the source database to update the latency.</li> <li>Note If you select an entire database as the object to synchronize, you can</li> </ul>
	create a heartbeat table. The heartbeat table is updated or receives data every second.

# 3.2. Precautions and limits for synchronizing data from a PolarDB for

## MySQL cluster

This topic describes the precautions and limits that you must take note of when you synchronize data from a PolarDB for MySQL cluster. To ensure that your data synchronization task runs as expected, you must read the precautions and limits before you configure the task.

## Scenarios of synchronizing data from a PolarDB for MySQL cluster

You can view the precautions and limits based on the following synchronization scenarios:

## ? Note

By default, Data Transmission Service (DTS) disables FOREIGN KEY constraints for the destination database in a data synchronization task. Therefore, the cascade and delete operations of the source database are not synchronized to the following types of destination databases:

- MySQL (ApsaraDB RDS for MySQL and self-managed MySQL dat abases)
- PolarDB for MySQL
- PolarDB-X 1.0
- AnalyticDB for MySQL
- AnalyticDB for PostgreSQL
- Elasticsearch
- Synchronize data between PolarDB for MySQL clusters
- Synchronize data from a PolarDB for MySQL cluster to an ApsaraDB RDS for MySQL instance or a selfmanaged MySQL database
- Synchronize data from a PolarDB for MySQL cluster to a PolarDB-X V1.0 instance
- Synchronize data from a PolarDB for MySQL cluster to an AnalyticDB for MySQL cluster
- Synchronize dat a from a PolarDB for MySQL cluster to an AnalyticDB for PostgreSQL instance
- Synchronize data from a PolarDB for MySQL cluster to a DataHub project
- Synchronize dat a from a PolarDB for MySQL cluster to an Elasticsearch cluster
- Synchronize data from a PolarDB for MySQL cluster to a Message Queue for Apache Kafka instance or a self-managed Kafka cluster
- Synchronize dat a from a PolarDB for MySQL cluster to an AnalyticDB for PostgreSQL instance

## Synchronize data between PolarDB for MySQL clusters

The following table describes the precautions and limits.

|--|--|

Category	Description
Limits on the source database	<ul> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to be synchronized and you need to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature must be enabled. The value of the loose_polar_log_bin parameter must be set to on. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> <li>If you perform only incremental data synchronization, the binary logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database must be stored for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In extreme cases, data may be inconsistent or lost. Make sure that you set the retention period of binary logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability and performance.</li> </ul>
Other limits	<ul> <li>Read-only nodes of the source cluster cannot be synchronized.</li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.</li> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination database. If you use tools other than DTS to write data to the destination database, we recommend that you do not use DMS to perform online DDL operations. Otherwise, data synchronization, database.</li> </ul>

Category	Description
	If you want to configure two-way data synchronization between PolarDB for MySQL clusters, take note of the following limits:
	• DTS supports two-way data synchronization only between two PolarDB for MySQL clusters. DTS does not support two-way data synchronization between multiple PolarDB for MySQL clusters.
Special	• Limits on DDL synchronization direction: To ensure data consistency and the stability of two-way data synchronization, you can synchronize DDL operations only in the forward direction.
cases	• When DTS runs a two-way data synchronization task, DTS creates a database named dts in the destination database to prevent circular synchronization. When the task is running, do not modify the dts database.
	• If the source or destination instance of a two-way data synchronization task resides in a region outside the Chinese mainland, data can be synchronized only within this region. Cross-region two-way synchronization is not supported. For example, if the source instance resides in the Japan (Tokyo) region, data can be synchronized only within the Japan (Tokyo) region and cannot be synchronized to the Germany (Frankfurt) region.

## Synchronize data from a PolarDB for MySQL cluster to an ApsaraDB RDS for MySQL instance or a self-managed MySQL database

The following table describes the precautions and limits.

Category

Category	Description
Limits on the source database	<ul> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to be synchronized and you need to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature must be enabled. The value of the loose_polar_log_bin parameter must be set to on. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> <li>If you perform only incremental data synchronization, the binary logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In extreme cases, data may be inconsistent or lost. Make sure that you set the retention period of binary logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability and performance.</li> </ul>
Ot her limits	<ul> <li>Read-only nodes of the source cluster cannot be synchronized.</li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.</li> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.</li> </ul>

## Synchronize data from a PolarDB for MySQL cluster to a PolarDB-X V1.0 instance

The following table describes the precautions and limits.

Category	Description
Limits on the source database	<ul> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to be synchronized and you need to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> </ul>
	<ul> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature must be enabled. The value of the loose_polar_log_bin parameter must be set to on. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	<ul> <li>If you perform only incremental data synchronization, the binary logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database must be stored for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In extreme cases, data may be inconsistent or lost. Make sure that you set the retention period of binary logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability and performance.</li> </ul>

#### Dat a Transmission Service

Category	Description
Other limits	<ul> <li>Requirements for the objects to be synchronized:</li> <li>DTS does not synchronize the following data types: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID_SNAPSHOT.</li> <li>Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to be synchronized. If the source database contains prefix indexes, data may fail to be synchronized.</li> <li>Read-only nodes of the source cluster cannot be synchronized.</li> <li>Schema synchronization is not supported. Before you configure a data synchronization task, you must create databases and tables in the destination instance.</li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables pace of the destination database is larger than that of the source database.</li> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination database.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and the destination database.<!--</td--></li></ul>

# Synchronize data from a PolarDB for MySQL cluster to an AnalyticDB for MySQL cluster

The following table describes the precautions and limits.

Category	Description
Limits on the source database	<ul> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to be synchronized and you need to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature must be enabled. The value of the loose_polar_log_bin parameter must be set to on. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> <li>If you perform only incremental data synchronization, the binary logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database must be stored for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In extreme cases, data may be inconsistent or lost. Make sure that you set the retention period of binary logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability and performance.</li> <li>During data synchronization, do not perform DDL operations to modify the primary key or add comments, for example, ALTER TABLE table_name COMMENT='Table comments';     The operations do not take effect.</li> </ul>

Category	Description
Other limits	<ul> <li>Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to be synchronized.</li> <li>Read-only nodes of the source cluster cannot be synchronized.</li> <li>Due to the limits of , if the disk space usage of the nodes in an cluster reaches 80%, the task is delayed and error messages are returned. We recommend that you estimate the required disk space based on the objects that you want to synchronize. You must ensure that the destination cluster has sufficient storage space.</li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.</li> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. If you use tools other than DTS to write data to the destination database.</li> </ul>

# Synchronize data from a PolarDB for MySQL cluster to an AnalyticDB for PostgreSQL instance

The following table describes the precautions and limits.
Category	Description	
Limits on the source database	<ul> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to be synchronized and you need to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature must be enabled. The value of the loose_polar_log_bin parameter must be set to on. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> <li>If you perform only incremental data synchronization, the binary logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database must be stored for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In extreme cases, data may be inconsistent or lost. Make sure that you set the retention period of binary logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability and perform DDL operations to modify the primary key or add comments, for example, ALTER TABLE table_name COMMENT='Table comments': The operations do not take effect.</li> </ul>	

Category	Description
Other limits	<ul> <li>Requirements for the objects to be synchronized:</li> <li>Only tables can be selected as the objects to be synchronized.</li> <li>DTS does not synchronize the following data types: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID_SINAPSHOT.</li> <li>Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to be synchronized.</li> <li>Read-only nodes of the source cluster cannot be synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the database surger to an that of the source database.</li> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination database.</li> </ul>

### Synchronize data from a PolarDB for MySQL cluster to a DataHub project

The following table describes the precautions and limits.

Category	Description
----------	-------------

Category	Description
Limits on the source dat <i>a</i> base	<ul> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to be synchronized and you need to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature must be enabled. The value of the loose_polar_log_bin parameter must be set to on. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> <li>The binary logs of the source database must be stored for more than 24 hours.</li> </ul>
Ot her limit s	<ul> <li>Full data synchronization is not supported. DTS does not synchronize historical data of the required objects from the source RDS instance to the destination DataHub instance.</li> <li>Only tables and databases can be selected as the objects to be synchronized.</li> <li>Read-only nodes of the source cluster cannot be synchronized.</li> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database when you use DMS to perform online DDL operations.</li> </ul>

### Synchronize data from a PolarDB for MySQL cluster to an Elasticsearch cluster

The following table describes the precautions and limits.

|--|--|--|

Category	Description
Limits on the source database	<ul> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to be synchronized and you need to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature must be enabled. The value of the loose_polar_log_bin parameter must be set to on. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> <li>If you perform only incremental data synchronization, the binary logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database must be stored for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In extreme cases, data may be inconsistent or lost. Make sure that you set the retention period of binary logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability and performance.</li> </ul>
Other limits	<ul> <li>Read-only nodes of the source cluster cannot be synchronized.</li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.</li> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables during data synchronize, you must perform the following steps: Modify the mappings of the table in the Elasticsearch cluster, perform DDL operations in the source MySQL database, and then pause and start the data synchronization task.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.</li> </ul>

#### Synchronize data from a PolarDB for MySQL cluster to a Message Queue for Apache Kafka instance or a self-managed Kafka cluster

The following table describes the precautions and limits.

Category	Description
Limits on the source database	<ul> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to be synchronized and you need to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature must be enabled. The value of the loose_polar_log_bin parameter must be set to on. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> <li>If you perform only incremental data synchronization, the binary logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database must be stored for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In extreme cases, data may be inconsistent or lost. Make sure that you set the retention period of binary logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability and performance.</li> </ul>
Ot her limit s	<ul> <li>Read-only nodes of the source cluster cannot be synchronized.</li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.</li> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on source tables during data synchronization. For more information, see Change schemas without locking tables.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. If you use tools other than DTS to write data to the destination database.</li> </ul>

# 3.3. Precautions and limits for synchronizing data from a PolarDB-XV2.0 instance

This topic describes the precautions and limits that you must take note of when you synchronize data from a instance. To ensure that your data synchronization task runs as expected, read the precautions and limits before you configure the task.

#### Scenarios of synchronizing data from a instance

You can view the precautions and limits based on the following synchronization scenarios:

#### ? Note

By default, Data Transmission Service (DTS) disables FOREIGN KEY constraints for the destination database in a data synchronization task. Therefore, the cascade and delete operations of the source database are not synchronized to the following destination databases:

- MySQL (ApsaraDB RDS for MySQL instances and self-managed MySQL databases)
- PolarDB for MySQL
- PolarDB-X V2.0
- AnalyticDB for MySQL
- AnalyticDB for PostgreSQL
- Elasticsearch
- Synchronize data from a PolarDB-X V2.0 instance to an ApsaraDB RDS for MySQL instance or a selfmanaged MySQL database
- Synchronize data from a PolarDB-X V2.0 instance to a PolarDB for MySQL cluster
- Synchronize data between PolarDB-X V2.0 instances
- Synchronize data from a PolarDB-X V2.0 instance to an AnalyticDB for MySQL cluster
- Synchronize dat a from a PolarDB-X V2.0 instance to a Message Queue for Apache Kafka instance
- Synchronize data from a PolarDB-X V2.0 instance to a DataHub project
- Synchronize dat a from a PolarDB-X V2.0 instance to an Elasticsearch cluster

### Synchronize data from a PolarDB-X V2.0 instance to an ApsaraDB RDS for MySQL instance or a self-managed MySQL database

Category

Category	Description
	• The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
	• If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.
	• The following requirements for binary logs must be met:
Limits on the source database	<ul> <li>The binary logging feature must be enabled in the PolarDB-X console. For more information, see Parameter settings. The value of the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	<ul> <li>If you perform only incremental data synchronization, the binary logs of the source database are retained for at least 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database are retained for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance.</li> </ul>
	• The instance must be compatible with MySQL V5.7.
	<ul> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> </ul>
Otherlinite	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.
other timits	• We recommend that you do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on source tables during data synchronization. Otherwise, data synchronization may fail.
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, we recommend that you do not use Data Management (DMS) to perform online DDL operations. Otherwise, data loss may occur in the destination database.

### Synchronize data from a PolarDB-X V2.0 instance to a PolarDB for MySQL cluster

Category	Description
	• The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
	• If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.
	• The following requirements for binary logs must be met:
Limits on the source database	<ul> <li>The binary logging feature must be enabled in the PolarDB-X console. For more information, see Parameter settings. The value of the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	<ul> <li>If you perform only incremental data synchronization, the binary logs of the source database are retained for at least 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database are retained for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance.</li> </ul>
	• The instance must be compatible with MySQL V5.7.
	• Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.
Otherlimite	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.
other timits	• We recommend that you do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on source tables during data synchronization. Otherwise, data synchronization may fail.
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, we recommend that you do not use Data Management (DMS) to perform online DDL operations. Otherwise, data loss may occur in the destination database.

#### Synchronize data between PolarDB-X V2.0 instances

Category	Description
	• The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
	• If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.
	• The following requirements for binary logs must be met:
Limits on the source database	<ul> <li>The binary logging feature must be enabled in the PolarDB-X console. For more information, see Parameter settings. The value of the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>
	<ul> <li>If you perform only incremental data synchronization, the binary logs of the source database are retained for at least 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database are retained for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance.</li> </ul>
	• The instance must be compatible with MySQL V5.7.
	• Schema synchronization is not supported in this scenario. Before you configure a data synchronization task, you must create databases and tables in the destination instance.
	• The destination instance must be compatible with MySQL 5.7.
	<ul> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> </ul>
Other limits	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.
	<ul> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on source tables during data synchronization. Otherwise, data synchronization may fail.</li> </ul>
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, we recommend that you do not use DMS to perform online DDL operations. Otherwise, data loss may occur in the destination database.

### Synchronize data from a PolarDB-X V2.0 instance to an AnalyticDB for MySQL cluster

Category	Description	
	• The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.	
	• If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.	
	• The following requirements for binary logs must be met:	
Limits on the source database	<ul> <li>The binary logging feature must be enabled in the PolarDB-X console. For more information, see Parameter settings. The value of the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>	
	<ul> <li>If you perform only incremental data synchronization, the binary logs of the source database are retained for at least 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database are retained for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance.</li> </ul>	
	• The instance must be compatible with MySQL V5.7.	

Description
• Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to be synchronized.
• Due to the limits of , if the disk space usage of the nodes in an cluster reaches 80%, the task is delayed and error messages are returned. We recommend that you estimate the required disk space based on the objects that you want to synchronize. You must ensure that the destination cluster has sufficient storage space.
• Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.
• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.
• We recommend that you do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on source tables during data synchronization. Otherwise, data synchronization may fail.
• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, we recommend that you do not use DMS to perform online DDL operations. Otherwise, data loss may occur in the destination database.

#### Synchronize data from a PolarDB-X V2.0 instance to a Message Queue for Apache Kafka instance

Category

Category	Description		
	• The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.		
	• If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.		
	• The following requirements for binary logs must be met:		
Limits on the source database	<ul> <li>The binary logging feature must be enabled in the PolarDB-X console. For more information, see Parameter settings. The value of the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>		
	<ul> <li>If you perform only incremental data synchronization, the binary logs of the source database are retained for at least 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database are retained for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance.</li> </ul>		
	• The instance must be compatible with MySQL V5.7.		

Category	Description			
	• Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.			
	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.			
Other limits	<ul> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on source tables during data synchronization. Otherwise, data synchronization may fail.</li> </ul>			
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, we recommend that you do not use DMS to perform online DDL operations. Otherwise, data loss may occur in the destination database.			
	• Only tables can be selected as the objects to be synchronized.			
	• DTS does not synchronize the data in a renamed table to the destination Kafka cluster. This applies if the new table name is not included in the objects to be synchronized. To synchronize the data in a renamed table to the destination Kafka cluster, you must <b>reselect the objects to be synchronized</b> . For more information, see Add an object to a data synchronization task.			

### Synchronize data from a PolarDB-X V2.0 instance to a DataHub project

Category

Category	Description		
	<ul> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the option of tables.</li> </ul>		
	The following requirements for binary logs must be met:		
Limits on the source database	<ul> <li>The following requirements for binary logs must be met:</li> <li>The binary logging feature must be enabled in the PolarDB-X console. For more information, see Parameter settings. The value of the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>		
	<ul> <li>If you perform only incremental data synchronization, the binary logs of the source database are retained for at least 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database are retained for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the SLA of DTS does not ensure service reliability and performance.</li> <li>The instance must be compatible with MySQL V5.7.</li> </ul>		
	<ul> <li>Limits on the objects to be synchronized;</li> </ul>		
	<ul> <li>Only tables can be selected as the objects to be synchronized.</li> </ul>		
	<ul> <li>After a data synchronization task is started, DTS does not synchronize columns that are created in the source database to the destination project.</li> </ul>		
Other limits	• We recommend that you do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on source tables during data synchronization. Otherwise, data synchronization may fail.		
	<ul> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, we recommend that you do not use DMS to perform online DDL operations. Otherwise, data loss may occur in the destination database.</li> <li>Full data synchronization is not supported. DTS does not synchronize the historical data in the source instance to the destination project.</li> </ul>		

### Synchronize data from a PolarDB-X V2.0 instance to an Elasticsearch cluster

Category	Description		
	• The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.		
	<ul> <li>If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> </ul>		
	• The following requirements for binary logs must be met:		
Limits on the source database	<ul> <li>The binary logging feature must be enabled in the PolarDB-X console. For more information, see Parameter settings. The value of the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>		
	<ul> <li>If you perform only incremental data synchronization, the binary logs of the source database are retained for at least 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database are retained for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance.</li> </ul>		
	• The instance must be compatible with MySQL V5.7.		

Category	Description		
	• DTS cannot be used to synchronize DDL operations. If a DDL operation is performed on a source table during data synchronization, you must remove the table from the data synchronization task. Then, you must remove the index that corresponds to the table from your cluster, and add the table to the data synchronization task again. For more information, see Remove an object from a data synchronization task and Add an object to a data synchronization task.		
	• If you want to add columns to a source table, modify the mappings of the index that corresponds to the table in the cluster. Then, perform DDL operations on the source table, pause the data synchronization task, and start the task again.		
Other limits	• Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.		
	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.		
	• We recommend that you do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on source tables during data synchronization. Otherwise, data synchronization may fail.		
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, we recommend that you do not use DMS to perform online DDL operations. Otherwise, data loss may occur in the destination database.		

### Synchronize data from a PolarDB-X V2.0 instance to a MaxCompute project

Category

Category	Description			
	• The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.			
	• If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.			
	• The following requirements for binary logs must be met:			
Limits on the source database	<ul> <li>The binary logging feature must be enabled in the PolarDB-X console. For more information, see Parameter settings. The value of the binlog_row_image parameter must be set to full. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>			
	<ul> <li>If you perform only incremental data synchronization, the binary logs of the source database are retained for at least 24 hours. If you perform both full data synchronization and incremental data synchronization, the binary logs of the source database are retained for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the binary logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of binary logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance.</li> </ul>			
	• The instance must be compatible with MySQL V5.7.			
	• Only tables can be selected as the objects to be synchronized.			
	<ul> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> </ul>			
	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.			
Other limits	<ul> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on source tables during data synchronization. Otherwise, data synchronization may fail.</li> </ul>			
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, we recommend that you do not use DMS to perform online DDL operations. Otherwise, data loss may occur in the destination database.			
	<ul> <li>MaxCompute does not support the PRIMARY KEY constraint. If network errors occur, DTS may synchronize duplicate data records to MaxCompute.</li> </ul>			

# 3.4. Limits for using PolarDB-X as the source instance

does not provide binary logs. To ensure data quality, we recommend that you take note of the limits when you perform business design, business development, and O&M changes.

#### Overview

- Limits related to business design
- Limits related to database architecture
- Limits related to O&M changes
- Potential risks related to data quality
- Suggestions to ensure data quality

#### Limits related to business design

- All tables must have primary keys. Otherwise, data inconsistency may occur (the destination database may contain duplicate data records).
- We recommend that you do not use the global secondary indexes (GSIs) of because they are updated asynchronously. If you use GSIs, DTS can guarantee only eventual consistency of data.
- The databases that you want to synchronize cannot be deployed in mixed mode (where the unit mode and the copy mode are mixed).

(?) Note In unit mode, users perform read and write operations in their respective unit nodes. Two-way synchronization is implemented between the databases in each unit node and those in the central node. In copy mode, users write data to the databases in the central node. The data is then synchronized to the databases in each unit node.

- If you use the underlying MySQL databases of to configure two-way data synchronization tasks, you must convert the Float and Double data types into Decimal for business tables. If you use as the source instance of a one-way data synchronization task, a data migration task, or a change tracking task, you do not need to convert the Float and Double data types.
- Data synchronization tasks, data migration tasks, and change tracking tasks do not support the following objects of : stored procedures, triggers, functions, views, and events.
- DTS does not support schema synchronization for . You must manually create objects such as databases and tables in the destination instance.
- The source instance must have sufficient capacity to support business growth.
- If MySQL databases of version 5.7 and 8.0 run on the instance, you cannot use the instance as the source instance of a change tracking task. You must configure a change tracking task for each MySQL database to track and consume data from the instance.

#### Limits related to database architecture

- The ApsaraDB RDS for MySQL instances used by a instance cannot be used by other instances.
- For data synchronization or migration between instances, the source and destination ApsaraDB RDS for MySQL instances must have equivalent deployment. For example, if the source instance uses four ApsaraDB RDS for MySQL instances, the destination instance must also use four ApsaraDB RDS for MySQL instances with the same specifications.

- The sharding rules of the source and destination instances must be the same. Otherwise, the data synchronization or migration task cannot be created.
- You can synchronize, migrate, or track the data of business tables in an instance. You cannot synchronize, migrate, or track the data of metadata tables or system tables in the instance.

#### Limits related to O&M changes

Category	Description	Impact and solution	
	Change a sharding rule, for example, change the shard key of a database or table or change the number of shards.	<ul> <li>Not supported. You must perform the following steps to recreate the task:</li> <li>1. Stop and delete the original DTS task.</li> <li>2. Wait until the changes are completed in the source database. Then, clear the data that has</li> </ul>	
	Change the number of instances at the storage layer, for example, scale out instances and migrate frequently-accessed tables.	<ul><li>been synchronized or migrated to the destination database.</li><li>3. Configure a data synchronization or migration task for each ApsaraDB RDS for MySQL instance in the instance.</li></ul>	
	Change the specifications of instances and switch workloads at the storage layer.	DTS tasks are not affected.	
	Change parameter settings.	The parameter settings of the source and destination databases must be the same. To change the parameter settings of instances at the storage layer, you must make sure that new parameters do not affect previous parameters.	
Storage layer		<b>Note</b> If you are not sure about the impact of changing parameter settings, you can contact technical support of Database Expert Service.	
	Change backup and recovery policies, and enable auditing and diagnostics for instances at the storage layer.	The change takes effect only on the current instance and does not affect other instances with replication relationships.	
DTS task	Perform DDL operations.	If you configure a DTS task to replicate data from multiple ApsaraDB RDS for MySQL instances in a instance to the destination database, performing DDL operations may cause task latency.	

Category	Description	Impact and solution
	Add tables.	<ul> <li>Not supported. You must perform the following steps:</li> <li>1. After you perform DDL operations to create tables in the destination database, perform the same DDL operations to create tables in the source database.</li> <li>2. Add the new tables to the objects of the data synchronization task.</li> <li>You can write data to the source database only after the preceding operations are completed. If you select tables as the objects of the data synchronization task, you must add the new tables to the source and destination instances at the storage layer.</li> </ul>
DDL operations at the database or	Add fields, add secondary indexes, delete indexes, and modify indexes (except for replacing secondary indexes with unique indexes).	<ul> <li>If you configure a DTS task based on a instance, you must perform DDL operations in the destination database and then perform the same DDL operations in the source database.</li> <li>If you configure a DTS task based on ApsaraDB RDS for MySQL instances attached to a instance, DTS automatically updates the operations. The following operations are supported: Add fields, add secondary indexes, delete indexes, and modify indexes (except for replacing secondary indexes with unique indexes).</li> </ul>
table level	Perform other DDL operations.	Only the preceding DDL operations are supported.

Category	Description	Impact and solution	
	Perform a switchover.	Before you perform a switchover, make sure that the DTS task is not delayed. Otherwise, data quality issues occur.	
	Perform a failover that meets the requirements of recovery point objective (RPO).	If a failure (such as network interruption, equipment failure, or Internet data center failure) occurs and the DTS task is delayed, you may need to perform a failover. In this case, if the difference between the time when the last data entry is updated to the destination database and the time when the failure occurs is less than the RPO, you can perform a failover to recover your business. For example, if the RPO is 5 minutes, the quality of the data within the 5 minutes cannot be guaranteed after you perform a failover. You may need to revise the data to ensure consistency.	
Switchover <b>Note</b> Switchover: After you use	<b>?</b> Note RPO represents the maximum amount of data that can be lost after a recovery from a failure. RPO is measured by time.		
DTS to synchronize or migrate data from the source database to the destination database, you switch workloads from the	Warning Failover: If the source instance or the data center where the source instance resides fails, you can switch workloads to a backup system. A failover is a lossy operation.		
source database to the destination database.	Perform a failover that does not meet the requirements of RPO.	The DTS task may be delayed because of the following reasons: a large number of DDL operations is performed in the source database, a network failure occurs, and the performance of the destination database is unfavorable. In this case, if the data center fails and the difference between the time when the last data entry is updated to the destination database and the time when the failure occurs is greater than the RPO, we recommend that you wait until the data center recovers before you perform a failover. For example, if the RPO is 5 minutes, the quality of the data within the 5 minutes cannot be guaranteed after you perform a failover. You may need to revise the data to ensure consistency.	

#### Potential risks related to data quality

Some changes or switchover operations may cause data quality issues such as schema inconsistency between the source and destination databases.

• If data latency occurs between the primary and secondary databases of the source instance, the data written to the primary database is not updated to the secondary database in a timely manner. In this case, if you perform a primary/secondary switchover in the source instance, DTS uses the secondary database of the source instance as the source database for data synchronization, data migration, or change tracking. As a result, the data that is not updated to the secondary database is

lost.

- If the DTS task is resumed from a network failure after you perform a switchover, DTS attempts to synchronize, migrate, or track the data generated before the failure occurs. This mechanism prevents data loss in the destination database. In this case, if the destination tables do not have primary keys, data will be inconsistent between the source and destination databases. If the destination tables have primary keys, data may not be consistent when DTS implements the retry mechanism, but data will remain consistent after the retry ends.
- The DTS task may be delayed due to network failures and DDL operations.
- The DTS task may be delayed or interrupted due to changes to the source database, unfavorable performance of the destination database, and schema inconsistency.

Alibaba Cloud cannot solve the preceding issues. You must recreate a DTS task or adjust the source and destination databases.

#### Suggestions to ensure data quality

- You must perform all DDL operations with caution. All DDL operations must be confirmed by the technical engineers to comply with the preceding limits.
- Do not directly perform DDL operations in your program code.

## 3.5. Solutions for using PolarDB-X as the source instance

Data Transmission Service (DTS) allows you to synchronize or migrate data from a instance to the destination database. However, if you directly use a instance as the source instance of a DTS task and more than two ApsaraDB RDS for MySQL instances are attached to the DRDS instance, performance bottlenecks and stability risks may occur. Your business may be affected. To achieve higher performance and stability, we recommend that you configure a task for each ApsaraDB RDS for MySQL instance.

#### Prerequisites

You have read the Limits for using PolarDB-X as the source instance and have made sure to follow the relevant conventions and precautions.

#### Solutions

Solutio n	Configuration method	Description
Solutio n 1	Configure a DTS task for each ApsaraDB RDS for MySQL instance that is attached to the instance. When you configure DTS tasks, you must map the database and table names of all the ApsaraDB RDS for MySQL instances to the database and table names of the destination instance. Compared with directly using the instance as the source instance of a DTS task, this solution ensures higher performance and stability.	We recommend that you use Solution 1, which provides higher performance and stability than Solution 2. In addition, the number of ApsaraDB RDS for MySQL instances attached to the instance is unlimited.
	Notice If you use this solution, you can select only tables as the objects to be synchronized.	

Solutio n	Configuration method	Description
Solutio n 2	Directly use the instance as the source instance to configure a data synchronization or migration task. If more than two ApsaraDB RDS for MySQL instances are attached to the DRDS instance, the stability and operability of the DTS task may be affected. Your business may also be affected.	If you use Solution 2 and more than two ApsaraDB RDS for MySQL instances are attached to the source instance, the performance and stability of the DTS task may be compromised.

#### Comparison between Solution 1 and Solution 2

ltem	Solution 1	Solution 2	
Performance	Multiple DTS tasks provide higher performance and support large amounts of data writes to the instance.	Only one DTS task is configured to migrate or synchronize data from the instance. When the business system writes large amounts of data to the source instance, performance bottlenecks occur.	
Stability	High. Multiple DTS tasks are configured to migrate or synchronize data from the ApsaraDB RDS for MySQL instances attached to the instance. If one of the DTS tasks fails, the other DTS tasks are not affected. You only need to recover the failed DTS task.	Medium. Only one DTS task is configured to migrate or synchronize data from the instance. If the DTS task fails, you must troubleshoot the entire task to resume data transmission.	
Ease of use	You must configure multiple DTS tasks. You must configure database and table name mapping for each task. You must map the database and table names of multiple ApsaraDB RDS for MySQL instances in the source instance to the names of the databases and tables in the destination instance.	You need to configure only one DTS task for the source instance.	
Resource usage	Multiple DTS instances are required.	Only one DTS instance is required.	

### How do the two solutions implement synchronization or incremental migration of DDL operations

DTS does not support synchronization or incremental migration of DDL operations from a instance. If DDL operations are performed in the source instance during data synchronization or migration, you can perform the following steps to ensure that data can be written to the destination database:

- 1. Release the DTS task.
- 2. Clear the destination database.
- 3. Reconfigure the task.

In some scenarios where is the source instance, you can synchronize or migrate DDL operations without releasing the DTS task. The following table describes specific scenarios and corresponding operations.

Scenario	Operation
You use Solution 1 and select tables as the objects to be synchronized.	<ul> <li>You can add tables to the objects that you select for the data synchronization task.</li> <li>You can add or remove columns only when you synchronize data between two instances. To do this, perform the following steps: <ol> <li>Add columns to or remove columns from the objects that you select for the data synchronization task.</li> <li>Add or remove columns in the destination database, and then perform the same operations in the source database. When DTS detects that a column already exists in the destination database, DTS ignores the error and does not display a write failure.</li> </ol> </li> </ul>
You use Solution 2 and select an entire database as the object to be migrated or synchronized.	<ul> <li>You can only add tables. To do this, you must add tables in the destination database and then perform the same operations in the source database.</li> <li>You cannot add or remove columns.</li> <li><b>Warning</b> If you add a column in the source instance, some physical tables at the underlying layer may contain the column whereas some physical tables do not contain the column. When DTS assembles the SQL statement, DTS may fail to find the column or lose the data in the column.</li> </ul>
You use Solution 2 and do not select an entire database as the object to be migrated or synchronized.	<ul> <li>You can only add tables. To do this, perform the following steps:</li> <li>i. Add tables to the objects that you select for the data synchronization task.</li> <li>ii. Add tables in the destination database, and then perform the same operations in the source database.</li> <li>You cannot add or remove columns.</li> </ul>

### 3.6. Precautions and limits for synchronizing data from an Oracle database

This topic describes the precautions and limits that you must take note of when you synchronize data from a self-managed Oracle database. To ensure that your data synchronization task runs as expected, you must read the precautions and limits before you configure the task.

#### Scenarios for synchronizing data from an Oracle database

You can view the precautions and limits based on the following synchronization scenarios:

#### ? Note

By default, Data Transmission Service (DTS) disables FOREIGN KEY constraints for the destination database in a data synchronization task. Therefore, the cascade and delete operations of the source database are not synchronized to AnalyticDB for PostgreSQL.

- Synchronize data from a self-managed Oracle database to an AnalyticDB for PostgreSQL instance
- Synchronize data from a self-managed Oracle database to a Message Queue for Apache Kafka instance or a self-managed Kafka cluster
- Synchronize dat a from a self-managed Oracle dat abase to a Dat aHub project

### Synchronize data from a self-managed Oracle database to an AnalyticDB for PostgreSQL instance

Category

Category	Description
Other limits	• DTS supports schema synchronization for the following types of objects: table, index, constraint, function, sequence, and view.
	<ul> <li>Warning         <ul> <li>Oracle and AnalyticDB for PostgreSQL are heterogeneous databases. DTS does not ensure that the schemas of the source and destination databases are consistent after schema synchronization. We recommend that you evaluate the impact of data type conversion on your business. For more information, see Data type mappings for schema synchronization.</li> <li>In this scenario, DTS is incompatible with triggers. We recommend that you delete the triggers of the source database to prevent data inconsistency caused by triggers. For more information, see Configure a data synchronization task for a source database that contains a trigger.</li> <li>For partitioned tables, DTS discards the partition definitions. You must define partitions in the destination database. We recommend that you synchronize data, evaluate the impact of data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> </ul> </li> <li>Before you synchronize dota, evaluate the impact of data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.</li> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DL operations on objects. Otherwise, data synchronization may fail.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronization latency may be inaccurate. If the latency of the source database for a long time, the synchronization latency may be inaccurate. If the latency of the synchronization task is too high, you can perform a DML operation on th</li></ul>
	<b>Note</b> If you select an entire database as the object to be synchronized, you can create a heartbeat table. The heartbeat table is updated or receives data every second.
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. For example, if you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use Data Management (DMS) to perform online DDL operations.

#### Synchronize data from a self-managed Oracle database to a Message Queue for Apache Kafka instance or a self-managed Kafka cluster

Lategory Description	
<ul> <li>Earegory</li> <li>Description</li> <li>Requirements for the objects to be</li> <li>The tables to be synchronized main and all fields must be unique. Oth duplicate data records.</li> <li>If the version of your Oracle datal be synchronized cannot exceed 3</li> <li>If you select tables as the objects tables (such as renaming tables or synchronize more than 1,000 table recommend that you split the tables, or configure a task to synchronize more than 1,000 table recommend that you split the tables, or configure a task to synchronize database when you configure the data tables, or configure a task to synchronized over Express Connect, you database when you configure the data synchronization task. After RAC database is not supported.</li> <li>The following requirements must but the redo logging and archive logs of the source database must perform both full data and incremental du logs of the source database must perform both full data and incremental cost of the source database must perform both full data and incrementation task may fail. In excel loss may occur. Make sure that you archive logs in accordance with the Service Level Agreement (SLA) of performance.</li> <li>If you perform a primary/secondary data synchronization task is running</li> </ul>	synchronized: ust have PRIMARY KEY or UNIQUE constraints, herwise, the destination database may contain base is 12c or later, the names of the tables to 0 bytes in length. s to be synchronized and you want to edit or columns), up to 1,000 tables can be chronization task. If you run a task to les, a request error occurs. In this case, we oles, configure multiple tasks to synchronize the chronize the entire database. Real Application Cluster (RAC) database tu must specify a virtual IP address (VIP) for the ata synchronization task. e is an Oracle RAC database, you can only use a s Name (SCAN) IP address when you configure ryou specify the VIP, node failover of the Oracle e met: ging features must be enabled. lata synchronization, the redo logs and archive t be stored for more than 24 hours. If you mental data synchronization, the redo logs and archive t be stored for at least seven days. completed, you can set the retention period to OTS may fail to obtain the redo logs and archive eptional circumstances, data inconsistency or ou set the retention period of redo logs and archive eptional circumstances, data inconsistency or ou set the retention period of redo logs and he preceding requirements. Otherwise, the DTS does not ensure service reliability and

Category	Description
Other limits	<ul> <li>DTS does not synchronize the data in a renamed table to the destination Kafka cluster. This applies if the new table name is not included in the objects to be synchronized. To synchronize the data in a renamed table to the destination Kafka cluster, you must reselect the objects to be synchronized. For more information, see Add an object to a data synchronization task.</li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation</li> </ul>
	in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.
	<ul> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on objects. Otherwise, data synchronization may fail.</li> </ul>
	• DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the latency of the synchronization task is too high, you can perform a DML operation on the source database to update the latency.
	<b>Note</b> If you select an entire database as the object to be synchronized, you can create a heartbeat table. The heartbeat table is updated or receives data every second.
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. For example, if you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use Data Management (DMS) to perform online DDL operations.

### Synchronize data from a self-managed Oracle database to a DataHub project

Category
----------

### 3.7. Precautions and limits for synchronizing data from a PostgreSQL database

This topic describes the precautions and limits that you must take note of when you synchronize data from a PostgreSQL database, such as a self-managed PostgreSQL database and an ApsaraDB RDS for PostgreSQL database. To ensure that your data synchronization task runs as expected, read the precautions and limits before you configure the task.

#### Scenarios of synchronizing data from a PostgreSQL database

You can view the precautions and limits based on the following synchronization scenarios:

#### ? Note

By default, Data Transmission Service (DTS) disables FOREIGN KEY constraints for the destination database in a data synchronization task. Therefore, the cascade and delete operations of the source database are not synchronized to the following destination databases:

- ApsaraDB RDS for PostgreSQL
- AnalyticDB for PostgreSQL
- PolarDB for Oracle
- ApsaraDB RDS for MySQL
- Synchronize data between PostgreSQL databases
- Synchronize data from a self-managed PostgreSQL database or an ApsaraDB RDS for PostgreSQL instance to an AnalyticDB for PostgreSQL instance
- Synchronize data from a self-managed PostgreSQL database to a PolarDB for Oracle cluster
- Synchronize data from an ApsaraDB RDS for PostgreSQL instance to an ApsaraDB RDS for MySQL instance

#### Synchronize data between PostgreSQL databases

Description

• Configure one-way data synchronization between ApsaraDB RDS for PostgreSQL instances

Category

Category	Description
Limits on the source database	<ul> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to be synchronized more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for write-ahead logging (WAL) logs must be met:</li> <li>The value of the wal_level parameter must be set to <i>logical</i>.</li> <li>If you perform only incremental data synchronization, the WAL logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the WAL logs of the source database must be stored for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the WAL logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of WAL logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability or performance.</li> <li>Limits on operations: If you want to perform a primary/secondary switchover on the source ApsaraDB RDS for PostgreSQL instance, the Logical Replication Slot Failover feature must be enabled. This prevents logical subscriptions from being interrupted and ensures that your data synchronization task can run as expected. For more information, see Logical Replication Slot Failover.</li> </ul>
	<ul> <li>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.</li> </ul>

Category	

Other limits Category	<ul> <li>During data synchronization, DTS creates a replication slot for the source Description database. The replication slot is prefixed with dts_sync_</li> <li>DTS automatically</li> </ul>
	clears historical replication slots every 90 minutes to reduce storage usage.
	<ul> <li>Note</li> <li>After the DTS instance is released, the replication slot is automatically deleted. If you modify the password of the source database or delete the IP address whitelist of DTS, the replication slot cannot be automatically deleted. In this case, you must delete the replication slot in the source database to prevent it from piling up.</li> <li>If the data synchronization task is released or fails, DTS automatically clears the replication slot. If a primary/secondary switchover is performed on the source ApsaraDB RDS for PostgreSQL instance, you must log on to the secondary database to clear the replication slot.</li> </ul>
	Query Editor Query History Scratch Pad
	<pre>1 SELECT * FROM pg_replication_slots;</pre>
	Data Output Explain Messages Notifications
	slot_name plugin aname slot_type datoid database text oid adatabase attraction aname boolean database active boolean database active boolean database boolean active boolean database boolean dat
	i <u>dts_sync_onu</u> pgoutput logical ib dtstestdata false true
	<ul> <li>synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. Therefore, after full data synchronization is completed, the size of the used tablespace of the destination database is larger than that of the source database.</li> </ul>
	<ul> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.</li> </ul>
	<ul> <li>If you use only DTS to write data to the destination database, you can use DMS to perform online DDL operations on source tables during data synchronization. For more information, see Change schemas without locking tables.</li> </ul>
	<ul> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use Data Management (DMS) to perform online DDL operations.</li> </ul>
	<ul> <li>If you perform both full data synchronization and incremental data synchronization and the tables to be synchronized in the source database contain foreign keys, triggers, or event triggers (PostgreSQL V11.5 and later), the session_replication_role parameter in the destination database must be set to replica. After the data synchronization task is released, you can change the value of the session_replication_role parameter back to origin.</li> </ul>

Category	Description
Special cases	If the source instance is an ApsaraDB RDS for PostgreSQL instance, take note of the following limits:
	RDS for PostgreSQL instance. Otherwise, the data synchronization task fails.

• Synchronize data from a self-managed PostgreSQL database to an ApsaraDB RDS for PostgreSQL instance

Category
Category	Description
Limits on the source database	<ul> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to be synchronized more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for write-ahead logging (WAL) logs must be met:</li> <li>The value of the wal_level parameter must be set to <i>logical</i>.</li> <li>If you perform only incremental data synchronization, the WAL logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the WAL logs of the source database must be stored for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the WAL logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of WAL logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability or performance.</li> <li>If you perform a primary/secondary switchover on a self-managed PostgreSQL database, the data synchronization task fails.</li> </ul>
	<ul> <li>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.</li> </ul>

Category	• If the schema is selected as the object to be synchronized, you must execute the Description ALTER TABLE schema.table REPLICA IDENTITY FULL; statement in the
Other limits	destination table before the data synchronization starts. This ensures data consistency. When you execute this statement, we recommend that you do not lock the table. Otherwise, a deadlock occurs.
	<ul> <li>Note</li> <li>Replace the schema and table in the preceding sample statement with the actual schema name and table name.</li> <li>We recommend that you perform this operation during off-peak hours.</li> </ul>
	<ul> <li>DTS does not check the validity of metadata such as sequences. You must manually check the validity of metadata.</li> <li>After your workloads are switched to the destination database, newly written sequences do not increment from the maximum value of the sequences in the source database. Therefore, you must query the maximum value of the sequences in the source database before you switch your workloads to the destination database. Then, you must specify the queried maximum value as the starting value of the sequences in the destination database. You can run the following statements to query the maximum value of the sequences in the source database:</li> </ul>
	<pre>do language plpgsql \$\$ declare   nsp name;   rel name;   val int8; begin   for nsp,rel in select nspname,relname from pg_class t2 ,   pg_namespace t3 where t2.relnamespace=t3.oid and t2.relkind='S'   loop     execute format(\$_\$select last_value from %I.%I\$_\$, nsp, rel) into val;     raise notice '%',     format(\$_\$select setval('%I.%I'::regclass, %s);\$_\$, nsp, rel, val+1);     end loop; end;     \$\$;</pre>
	<ul> <li>DTS creates the following temporary tables in the source database to obtain the DDL statements of incremental data, the schemas of incremental tables, and the heartbeat information. During data synchronization, do not delete temporary tables in the source database. Otherwise, exceptions occur. After the DTS instance is released, temporary tables are automatically deleted.         <pre>public.DTS_PG_CLASS , public.DTS_PG_ATTRIBUTE , public.DTS_PG_TY PE , public.DTS_PG_ENUM , public.DTS_POSTGRES_HEARTBEAT , public .DTS_DDL_COMMAND , and public.DTS_ARGS_SESSION .</pre> </li></ul> O To ensure that the latency of data synchronization is accurate, DTS adds a         heartbeat table to the source database. The name of the heartbeat table is dts_         postgres_heartbeat.

Category	<ul> <li>During data synchronization, DTS creates a replication slot for the source Description Gatabase. The replication slot is prefixed with <a href="https://dts_sync_">dts_sync_</a>. DTS automatically</li> </ul>
	clears historical replication slots every 90 minutes to reduce storage usage.
	<b>Note</b> If the data synchronization task is released or fails, DTS automatically clears the replication slot. If a primary/secondary switchover is performed on the source ApsaraDB RDS for PostgreSQL instance, you must log on to the secondary database to clear the replication slot.
	Query Editor         Query History         Scratch Pad         X
	1 SELECT * FROM pg_replication_slots;
	Data Output Explain Messages Notifications
	slot_name plugin a slot_type a datoid a database temporary a active a active_pid active_pid integer a
	i <u>dts_sync_onu</u> pgoutput logical io dtstestdata faise true
	Service you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.
	<ul> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. Therefore, after full data synchronization is completed, the size of the used tablespace of the destination database is larger than that of the source database.</li> </ul>
	<ul> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.</li> </ul>
	<ul> <li>If you use only DTS to write data to the destination database, you can use DMS to perform online DDL operations on source tables during data synchronization. For more information, see Change schemas without locking tables.</li> </ul>
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database
	when you use DMS to perform online DDL operations.
configure two-way	and a symptometry and the tables to be supervertised in the source database contain

 Category
 synchronization and the tables to be synchronized in the source database contain

 Deforipitionkeys, triggers, or event triggers (PostgreSQL V11.5 and later), the

 session\_replication\_role parameter in the destination database must be set to

 replica. After the data synchronization task is released, you can change the value

 of the session\_replication\_role parameter back to origin.

Category	Description
	• The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
	<ul> <li>If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to be synchronized more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> </ul>
	• The following requirements for write-ahead logging (WAL) logs must be met:
	The value of the wal_level parameter must be set to <i>logical</i> .
Limits on the source and destination databases	<ul> <li>If you perform only incremental data synchronization, the WAL logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the WAL logs of the source database must be stored for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the WAL logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of WAL logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability or performance.</li> <li>If you want to perform a primary/secondary switchover on the source ApsaraDB RDS for PostgreSQL instance, the Logical Replication Slot Failover feature must be enabled. This prevents logical subscriptions from being interrupted and ensures that your data synchronization task can run as expected. For more information, see Logical Replication Slot Failover.</li> </ul>
	<ul> <li>You can configure data synchronization tasks only in the China (Hangzhou), China (Shanghai), and China (Qingdao) region.</li> </ul>
	<ul> <li>A single data synchronization task can synchronize data from only one database.</li> <li>To synchronize data from multiple databases, you must create a data synchronization task for each database.</li> </ul>
	• If the schema is selected as the object to be synchronized, you must execute the ALTER TABLE schema.table REPLICA IDENTITY FULL; statement in the destination table before the data synchronization starts. This ensures data consistency. When you execute this statement, we recommend that you do not lock the table. Otherwise, a deadlock occurs.
	② Note
	<ul> <li>Replace the schema and table in the preceding sample statement with the actual schema name and table name.</li> </ul>
	<ul> <li>We recommend that you perform this operation during off-peak hours.</li> </ul>
	<ul> <li>DTS does not check the validity of metadata such as sequences. You must manually check the validity of metadata.</li> </ul>

Category	<ul> <li>After your workloads are switched to the destination database, newly written Description sequences do not increment from the maximum value of the sequences in the</li> </ul>
	source database. Therefore, you must query the maximum value of the sequences in the source database before you switch your workloads to the destination database. Then, you must specify the queried maximum value as the starting value of the sequences in the destination database. You can run the following statements to query the maximum value of the sequences in the source database:
	<pre>do language plpgsql \$\$ declare     nsp name;     rel name;     val int8; begin     for nsp,rel in select nspname,relname from pg_class t2 ,     pg_namespace t3 where t2.relnamespace=t3.oid and t2.relkind='S'     loop     execute format(\$_\$select last_value from %I.%I\$_\$, nsp, rel)     into val;     raise notice '%',     format(\$_\$select setval('%I.%I'::regclass, %s);\$_\$, nsp, rel,     val+1);     end loop; end;     \$\$; </pre>
	<pre>instance is released, temporary tables are automatically deleted.     public.DTS_PG_CLASS , public.DTS_PG_ATTRIBUTE , public.DTS_PG_TY PE , public.DTS_PG_ENUM , public.DTS_POSTGRES_HEARTBEAT , public .DTS_DDL_COMMAND , and public.DTS_ARGS_SESSION .</pre>
	• To ensure that the latency of data synchronization is accurate, DTS adds a heartbeat table to the source database. The name of the heartbeat table is <i>dts_postgres_heartbeat</i> .
Other limits	

Category	<ul> <li>During data synchronization, DTS creates a replication slot for the source Description database. The replication slot is prefixed with dts_sync DTS automatically</li> </ul>
	clears historical replication slots every 90 minutes to reduce storage usage.
	<ul> <li>Note</li> <li>After the DTS instance is released, the replication slot is automatically deleted. If you modify the password of the source database or delete the IP address whitelist of DTS, the replication slot cannot be automatically deleted. In this case, you must delete the replication slot in the source database to prevent it from piling up.</li> <li>If the data synchronization task is released or fails, DTS automatically clears the replication slot. If a primary/secondary switchover is performed on the source ApsaraDB RDS for PostgreSQL instance, you must log on to the secondary database to clear the replication slot.</li> </ul>
	Query Editor         Query History         Scratch Pad         x
	<pre>1 SELECT * FROM pg_replication_slots;</pre>
	Data Output Explain Messages Notifications
	active_pid name plugin name biot_type text database database active_pid active_pid integer
	1 dts_sync_ohu pgoutput logical 16 dtstestdata false true
	<ul> <li>performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. Therefore, after full data synchronization is completed, the size of the used tablespace of the destination database.</li> <li>We recommend that you do not use gh-ost or pt-online-schema-change to</li> </ul>
	perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.
	<ul> <li>If you use only DTS to write data to the destination database, you can use DMS to perform online DDL operations on source tables during data synchronization. For more information, see Change schemas without locking tables.</li> </ul>
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.
	<ul> <li>If you perform both full data synchronization and incremental data synchronization and the tables to be synchronized in the source database contain foreign keys, triggers, or event triggers (PostgreSQL V11.5 and later), the session_replication_role parameter in the destination database must be set to replica. After the data synchronization task is released, you can change the value of the session_replication_role parameter back to origin.</li> </ul>

Category	Description
Special cases	If the source instance is an ApsaraDB RDS for PostgreSQL instance, take note of the following limits:
	During data synchronization, do not modify the endpoint and zone of the ApsaraDB RDS for PostgreSQL instance. Otherwise, the data synchronization task fails.

## Synchronize data from a self-managed PostgreSQL database or an ApsaraDB RDS for PostgreSQL instance to an AnalyticDB for PostgreSQL instance

Category	Description
Limits on the source database	<ul> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to be synchronized more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for write-ahead logging (WAL) logs must be met:</li> <li>The value of the wal_level parameter must be set to <i>logical</i>.</li> <li>If you perform only incremental data synchronization, the WAL logs of the source database must be stored for at least seven days. After full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the WAL logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of WAL logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability or perform a primary/secondary switchover on a self-managed PostgreSQL database, the data synchronization task fails.</li> <li>If you perform a primary/secondary switchover on the source ApsaraDB RDS for PostgreSQL instance, the Logical Replication Slot Failover feature must be enabled. This prevents logical subscriptions from being interrupted and ensures that your data synchronization task can run as expected. For more information, see Logical Replication Slot Failover.</li> </ul>

Category	Description
	<ul> <li>After the DTS instance is released, the replication slot is automatically deleted. If you modify the password of the source database or delete the IP address whitelist of DTS, the replication slot cannot be automatically deleted. In this case, you must delete the replication slot in the source database to prevent it from piling up.</li> <li>If the data synchronization task is released or fails, DTS automatically clears the replication slot. If a primary/secondary switchover is performed on the source ApsaraDB RDS for PostgreSQL instance, you must log on to the secondary database to clear the replication slot.</li> </ul>
	Query Editor     Query History     Scratch Pad       1     SELECT * FROM pg_replication_slots;
	Date Output Europein Massage Matifications
	slot_name plugin slot_type datoid database temporary active pid integer
	1 dts_sync_ohu pgoutput logical 16 dtstestdata false true
	<ul> <li>performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. Therefore, after full data synchronization is completed, the size of the used tablespace of the destination database is larger than that of the source database.</li> </ul>
	perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.
	<ul> <li>If you use only DTS to write data to the destination database, you can use DMS to perform online DDL operations on source tables during data synchronization. For more information, see Change schemas without locking tables.</li> </ul>
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.
Special cases	<ul> <li>If the source instance is an ApsaraDB RDS for PostgreSQL instance, take note of the following limits:</li> <li>During data synchronization, do not modify the endpoint and zone of the ApsaraDB RDS for PostgreSQL instance. Otherwise, the data synchronization task fails.</li> </ul>

## Synchronize data from a self-managed PostgreSQL database to a PolarDB for Oracle cluster

Category	Description
Limits on the source database	<ul> <li>If you perform a primary/secondary switchover on a self-managed PostgreSQL database, the data synchronization task fails.</li> </ul>
	<ul> <li>Schema synchronization is not supported. Before you configure a data synchronization task, you must create databases and tables in the destination instance.</li> <li>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.</li> <li>If the schema is selected as the object to be synchronized, you must execute the ALTER TABLE schema.table REPLICA IDENTITY FULL; statement in the destination table before the data synchronization starts. This ensures data consistency. When you execute this statement, we recommend that you do not lock the table. Otherwise, a deadlock occurs.</li> </ul>
	<ul> <li>Note</li> <li>Replace the schema and table in the preceding sample statement with the actual schema name and table name.</li> <li>We recommend that you perform this operation during off-peak hours.</li> </ul>
	<ul> <li>DTS does not check the validity of metadata such as sequences. You must manually check the validity of metadata.</li> <li>After your workloads are switched to the destination database, newly written sequences do not increment from the maximum value of the sequences in the source database. Therefore, you must query the maximum value of the sequences in the source database before you switch your workloads to the destination database. Then, you must specify the queried maximum value as the starting value of the sequences in the destination database. You can run the following statements to query the maximum value of the sequences in the source database:</li> </ul>

Category	Deschptionguage plpgsql \$\$
Other limits	<pre>nsp name; rel name; val int8; begin for nsp,rel in select nspname,relname from pg_class t2 , pg_namespace t3 where t2.relnamespace=t3.oid and t2.relkind='S' loop execute format(\$_\$select last_value from %I.%I\$_\$, nsp, rel) into val; raise notice '%', format(\$_\$select setval('%I.%I'::regclass, %s);\$_\$, nsp, rel, val+1); end loop; end; \$\$;</pre>
	<ul> <li>DTS creates the following temporary tables in the source database to obtain the DDL statements of incremental data, the schemas of incremental tables, and the heartbeat information. During data synchronization, do not delete temporary tables in the source database. Otherwise, exceptions occur. After the DTS instance is released, temporary tables are automatically deleted.         <pre>public.DTS_PG_CLASS , public.DTS_PG_ATTRIBUTE , public.DTS_PG_TYP</pre> <pre>public.DTS_PG_ENUM , public.DTS_POSTGRES_HEARTBEAT , public.DT</pre> <pre>public.OMMAND , and public.DTS_ARGS_SESSION .</pre> </li> <li>To ensure that the latency of data synchronization is accurate, DTS adds a heartbeat table to the source database. The name of the heartbeat table is <i>dts_po</i></li></ul>
	<ul> <li>During data synchronization, DTS creates a replication slot for the source database. The replication slot is prefixed with dts_sync DTS automatically clears historical replication slots every 90 minutes to reduce storage usage.</li> </ul>
	<b>Note</b> If the data synchronization task is released or fails, DTS automatically clears the replication slot. If a primary/secondary switchover is performed on the source ApsaraDB RDS for PostgreSQL instance, you must log on to the secondary database to clear the replication slot.
	Query Editor     Query History     Scratch Pad       1     SELECT * FROM pg_replication_slots;
	Data Output     Explain     Messages     Notifications       slot_name     aname     plugin     slot_type     datoid     database     temporary     active_pid     active_pid       1     dts_sync_ohu     pgoutput     logical     16     dtstestdata     false     true
	<ul> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> </ul>

• During full data synchronization, concurrent INSERT operations cause fragmentation Synchronize data from the synchronization of the prost of the prost of the destination of the destination of the destination database is larger an ApsaraDB RDS for MySOL singtance database.

Category	Description
Limits on the source database	<ul> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to be synchronized more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements for write-ahead logging (WAL) logs must be met:</li> <li>The value of the wal_level parameter must be set to <i>logical</i>.</li> <li>If you perform only incremental data synchronization, the WAL logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the WAL logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of WAL logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability or perform a primary/secondary switchover on the source ApsaraDB RDS for PostgreSQL instance, the Logical Replication Slot Failover feature must be enabled. This prevents logical subscriptions from being interrupted and ensures that your data synchronization task can run as expected. For more information, see Logical Replication Slot Failover.</li> </ul>
	<ul> <li>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.</li> <li>If the schema is selected as the object to be synchronized, you must execute the ALTER TABLE schema.table REPLICA IDENTITY FULL; statement in the destination table before the data synchronization starts. This ensures data consistency. When you execute this statement, we recommend that you do not lock the table. Otherwise, a deadlock occurs.</li> <li>Note <ul> <li>Replace the schema and table in the preceding sample statement with the actual schema name and table name.</li> <li>We recommend that you perform this operation during off-peak hours.</li> </ul> </li> <li>DTS creates the following temporary tables in the source database to obtain the pDL statements of incremental data, the schemas of incremental tables, and the heartbeat information. During data synchronization, do not delete temporary tables in the source database. The source database. The source database. The source database. To be synchronization, do not delete temporary tables in the source database. The sum of the source database. The source database is released, temporary tables are automatically deleted.</li> </ul>

1	
Category	<pre>public.DTS_PG_CLASS , public.DTS_PG_ATTRIBUTE , public.DTS_PG_TYP Description E , public.DTS_PG_ENUM , public.DTS_POSTGRES_HEARTBEAT , public.DT</pre>
	S_DDL_COMMAND , and public.DTS_ARGS_SESSION .
	• To ensure that the latency of data synchronization is accurate, DTS adds a heartbeat table to the source database. The name of the heartbeat table is <i>dts_po stgres_heartbeat</i> .
	• During data synchronization, DTS creates a replication slot for the source database. The replication slot is prefixed with <a href="https://dts_sync_">dts_sync_</a> . DTS automatically clears
	historical replication slots every 90 minutes to reduce storage usage.
	⑦ Note
Other limits	<ul> <li>After the DTS instance is released, the replication slot is automatically deleted. If you modify the password of the source database or delete</li> </ul>
	the IP address whitelist of DTS, the replication slot cannot be automatically deleted. In this case, you must delete the replication slot
	In the source database to prevent it from plung up.
	<ul> <li>If the data synchronization task is released or fails, DIS automatically clears the replication slot. If a primary/secondary switchover is</li> </ul>
	performed on the source ApsaraDB RDS for PostgreSQL instance, you
	must log on to the secondary database to clear the replication slot.
	Ourse Filling Ourse Mitcherg
	1     SELECT * FROM pg_replication_slots;
	Data Output Explain Messages Notifications
	slot_name and plugin and slot_type and database and temporary active active boolean active.pid integer
	1 ots_sync_onu pgoutput logical 16 otstestdata faise true
	<ul> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> </ul>
	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. Therefore, after full data synchronization
	is completed, the size of the used tablespace of the destination database is larger than that of the source database.
	• We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.
	<ul> <li>If you use only DTS to write data to the destination database, you can use DMS to perform online DDL operations on source tables during data synchronization. For more information, see Change schemas without locking tables.</li> </ul>
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you
	use DMS to perform online DDL operations. If the source instance is an ApsaraDB RDS for PostgreSQL instance, take note of the following limits:
Special cases	Tollowing limits: During data synchronization, do not modify the endpoint and zone of the ApsaraDB RDS for PostgreSQL instance. Otherwise, the data synchronization task fails.

## 3.8. Precautions and limits for synchronizing data from an SQL Server database

This topic describes the precautions and limits when you synchronize data from an SQL Server database, such as a self-managed SQL Server database and an database. To ensure that your data synchronization task runs as expected, you must read the precautions and limits before you configure the task.

#### Scenarios of synchronizing data from an SQL Server database

You can click one of the following synchronization scenarios to view its precautions and limits.

#### ? Note

By default, Data Transmission Service (DTS) disables foreign key constraints for the destination database in a data synchronization task. Therefore, the cascade and delete operations of the source database are not synchronized to the following destination databases:

- SQL Server
- MySQL (ApsaraDB RDS for MySQL and self-managed MySQL dat abases)
- PolarDB MySQL
- AnalyticDB for MySQL
- AnalyticDB for PostgreSQL
- Synchronize data between SQL Server databases
- Synchronize data from an ApsaraDB RDS for SQL Server instance to a MySQL database
- Synchronize data from an SQL Server database to a PolarDB for MySQL cluster
- Synchronize dat a from an SQL Server dat abase to an AnalyticDB for MySQL cluster
- Synchronize dat a from an SQL Server dat abase to an AnalyticDB for Post greSQL instance

#### Synchronize data between SQL Server databases

The following table describes the precautions and limits.

Category
----------

Category	Description
Limits on the source database	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>A single data synchronization task can synchronize up to 10 databases. If you want to synchronize more than 10 databases, we recommend that you split the tables to synchronize or configure multiple tasks to synchronize the database. Otherwise, the performance and stability of your data synchronization task may be compromised.</li> <li>The following requirements for data logs must be met:</li> <li>The data logging feature must be enabled. The backup mode must be set to Full, and full logical backup must be performed.</li> <li>If you perform only incremental data synchronization, the data logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the data logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of data logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance.</li> </ul>

Category	Description
	<ul> <li>DTS does not synchronize data of the following types: CURSOR, ROWVERSION, SQL_VARIANT, HIERACHYID, and GEOMETRY.</li> <li>If you synchronize data between different versions of databases, make sure that</li> </ul>
	the database versions are compatible.
	<ul> <li>If you set SQL Server Incremental Synchronization Mode to Incremental Synchronization Based on Logs of Source Database in the Configure Objects and Advanced Settings step, the tables to synchronize must have clustered indexes that contain primary key columns. In addition, the table to synchronize cannot be compressed tables and cannot contain computed columns. Ignore the preceding limits in the mixed log-based parsing mode.</li> </ul>
	<ul> <li>In the Incremental Synchronization Based on Logs of Source Database mode, DTS adds a heartbeat table named <i>dts_log_heart_beat</i> to the source database to ensure that the latency of data synchronization is accurate. In the mixed log-based parsing incremental synchronization mode, DTS creates a trigger named <i>dts_cdc_sy nc_ddl</i>, a heartbeat table named <i>dts_sync_progress</i>, and a storage table named <i>dt s_cdc_ddl_history</i> by using DDL and enables Change Data Capture (CDC) for the database and some tables.</li> </ul>
Other limits	• Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.
	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. Therefore, after the full data synchronization is complete, the tablespace of the destination database is larger than that of the source database.
	• We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use Data Management (DMS) to perform online DDL operations.

### Synchronize data from an ApsaraDB RDS for SQL Server instance to a MySQL database

The following table describes the precautions and limits when you synchronize data from an ApsaraDB RDS for SQL Server instance to a MySQL database, such as a self-managed MySQL database or an ApsaraDB RDS for MySQL database.

Category

Category	Description
	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> </ul>
Limits on the	• A single data synchronization task can synchronize up to 10 databases. If you want to synchronize more than 10 databases, we recommend that you split the tables to synchronize or configure multiple tasks to synchronize the databases. Otherwise, the performance and stability of your data synchronization task may be compromised.
source database	<ul> <li>The following requirements for data logs must be met:</li> </ul>
	<ul> <li>The data logging feature must be enabled. The backup mode must be set to Full, and full logical backup must be performed.</li> </ul>
	<ul> <li>If you perform only incremental data synchronization, the data logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the data logs of the source database must be stored for at least seven days. After the full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the data logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of data logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance.</li> </ul>

Category	Description
	<ul> <li>Requirements for the objects to synchronize:</li> <li>DTS supports initial schema synchronization for the following types of objects: database, schema, and table.</li> <li>DTS does not synchronize data of the following types: TIMESTAMP, CURSOR, ROWVERSION, HIERACHYID, SQL_VARIANT, SPATIAL GEOMETRY, SPATIAL GEOGRAPHY, and TABLE.</li> </ul>
	• If you set SQL Server Incremental Synchronization Mode to Incremental Synchronization Based on Logs of Source Database in the Configure Objects and Advanced Settings step, the tables to synchronize must have clustered indexes that contain primary key columns. In addition, the table to synchronize cannot be compressed tables and cannot contain computed columns. Ignore the preceding limits in the mixed log-based parsing mode.
	Complex synchronous DDL operations are not supported at present.
Other limits	• In the Incremental Synchronization Based on Logs of Source Database mode, DTS adds a heartbeat table named <i>dts_log_heart_beat</i> to the source database to ensure that the latency of data synchronization is accurate. In the mixed log-based parsing incremental synchronization mode, DTS creates a trigger named <i>dts_cdc_sy nc_ddl</i> , a heartbeat table named <i>dts_sync_progress</i> , and a storage table named <i>dt s_cdc_ddl_history</i> by using DDL and enables Change Data Capture (CDC) for the database and some tables.
	• Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.
	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. Therefore, after the full data synchronization is complete, the tablespace of the destination database is larger than that of the source database.
	• We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.

## Synchronize data from an SQL Server database to a PolarDB for MySQL cluster

The following table describes the precautions and limits.

Category

Category	Description
Limits on the source database	• • • •
Other limits	<ul> <li>Requirements for the objects to synchronize:</li> <li>DTS supports initial schema synchronization for the following types of objects: database, schema, and table.</li> <li>DTS does not synchronize data of the following types: TIMESTAMP, CURSOR, ROWVERSION, HIERACHYID, SQL_VARIANT, SPATIAL GEOMETRY, SPATIAL GEOGRAPHY, and TABLE.</li> <li>Complex synchronous DDL operations are not supported at present.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination database. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.</li> </ul>

### Synchronize data from an SQL Server database to an AnalyticDB for MySQL cluster

Category

Category	Description
Limits on the source database	<ul> <li>The tables to synchronize must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select tables as the objects to synchronize and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>A single data synchronization task can synchronize up to 10 databases. If you want to synchronize more than 10 databases, we recommend that you split the tables to synchronize more than 10 databases, we recommend that you split the tables to synchronize or configure multiple tasks to synchronization task may be compromised.</li> <li>The following requirements for data logs must be met:</li> <li>The data logging feature must be enabled. The backup mode must be set to Full, and full logical backup must be performed.</li> <li>If you perform only incremental data synchronization, the data logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the data logs of the source database must be stored for at least seven days. After the full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the data logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of data logs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not more than 24 hours.</li> </ul>
	<ul> <li>Requirements for the objects to synchronize:</li> <li>DTS supports initial schema synchronization for the following types of objects: schema, table, view, function, and procedure.</li> </ul>
	warning ApsaraDB RDS for SQL Server and AnalyticDB for PostgreSQL are heterogeneous databases. The data types that they support do not have one-to-one correspondence. We recommend that you evaluate the impact of data type conversion on your business. For more information, see Data type mappings for schema synchronization.
	<ul> <li>DTS does not synchronize the schemas of assemblies, service brokers, full-text indexes, full-text catalogs, distributed schemas, distributed functions, CLR stored procedures, CLR scalar-valued functions, CLR table-valued functions, internal tables, systems, or aggregate functions.</li> </ul>
	<ul> <li>DTS does not synchronize data of the following types: TIMESTAMP, CURSOR, ROWVERSION, HIERACHYID, SQL_VARIANT, SPATIAL GEOMETRY, SPATIAL GEOGRAPHY, and TABLE.</li> </ul>
	• DTS does not synchronize tables that contain computed columns.
	<ul> <li>If you set SQL Server Incremental Synchronization Mode to Incremental Synchronization Based on Logs of Source Database in the Configure Objects and Advanced Settings step, the tables to synchronize must have</li> </ul>

Category	clustered indexes that contain primary key columns. In addition, the table to Description synchronize cannot be compressed tables and cannot contain computed columns.
	Ignore the preceding limits in the mixed log-based parsing mode.
Other limits	<ul> <li>In the Incremental Synchronization Based on Logs of Source Database mode, DTS adds a heartbeat table named <i>dts_log_heart_beat</i> to the source database to ensure that the latency of data synchronization is accurate. In the mixed log-based parsing incremental synchronization mode, DTS creates a trigger named <i>dts_cdc_sy nc_ddl</i>, a heartbeat table named <i>dts_sync_progress</i>, and a storage table named <i>dt s_cdc_ddl_history</i> by using DDL and enables Change Data Capture (CDC) for the database and some tables.</li> </ul>
	• Due to the limits of , if the disk space usage of the nodes in an cluster reaches 80%, the performance of data writing to the destination database is compromised and the DTS task is delayed. If the usage reaches 90%, data cannot be written to the destination database and error messages are returned. We recommend that you estimate the required disk space based on the objects that you want to synchronize. You must ensure that the destination cluster has sufficient storage space.
	• Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.
	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. Therefore, after the full data synchronization is complete, the tablespace of the destination database is larger than that of the source database.
	• We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.

## Synchronize data from an SQL Server database to an AnalyticDB for PostgreSQL instance

The following table describes the precautions and limits.

y
---

Category	Description		
	<ul> <li>Requirements for the objects to synchronize:</li> <li>DTS supports initial schema synchronization for the following types of objects: schema, table, view, function, and procedure.</li> </ul>		
	• warning ApsaraDB RDS for SQL Server and AnalyticDB for PostgreSQL are heterogeneous databases. The data types that they support do not have one-to-one correspondence. We recommend that you evaluate the impact of data type conversion on your business. For more information, see Data type mappings for schema synchronization.		
	<ul> <li>DTS does not synchronize the schemas of assemblies, service brokers, full-text indexes, full-text catalogs, distributed schemas, distributed functions, CLR stored procedures, CLR scalar-valued functions, CLR table-valued functions, internal tables, systems, or aggregate functions.</li> </ul>		
	<ul> <li>DTS does not synchronize data of the following types: TIMESTAMP, CURSOR, ROWVERSION, HIERACHYID, SQL_VARIANT, SPATIAL GEOMETRY, SPATIAL GEOGRAPHY, and TABLE.</li> </ul>		
	• DTS does not synchronize tables that contain computed columns.		
Ot her limits	<ul> <li>If you set SQL Server Incremental Synchronization Mode to Incremental Synchronization Based on Logs of Source Database in the Configure Objects and Advanced Settings step, the tables to synchronize must have clustered indexes that contain primary key columns. In addition, the table to synchronize cannot be compressed tables and cannot contain computed columns. Ignore the preceding limits in the mixed log-based parsing mode.</li> </ul>		
	• In the Incremental Synchronization Based on Logs of Source Database mode, DTS adds a heartbeat table named <i>dts_log_heart_beat</i> to the source database to ensure that the latency of data synchronization is accurate. In the mixed log-based parsing incremental synchronization mode, DTS creates a trigger named <i>dts_cdc_sy nc_ddl</i> , a heartbeat table named <i>dts_sync_progress</i> , and a storage table named <i>dt s_cdc_ddl_history</i> by using DDL and enables Change Data Capture (CDC) for the database and some tables.		
	• Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.		
	• During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. Therefore, after the full data synchronization is complete, the tablespace of the destination database is larger than that of the source database.		
	• We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on source tables during data synchronization. Otherwise, data synchronization may fail.		
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.		

## 3.9. Precautions and limits for synchronizing data from a MongoDB database

This topic describes the precautions and limits when you synchronize data from a MongoDB database, such as a self-managed MongoDB database and an ApsaraDB for MongoDB instance. To ensure that your data synchronization task runs as expected, you must read the precautions and limits before you configure the task.

#### Scenarios of synchronizing data from a MongoDB database

You can click one of the following synchronization scenarios to view its precautions and limits:

- Synchronize data from a ApsaraDB for MongoDB instance (replica set architecture) to another ApsaraDB for MongoDB instance (replica set architecture or sharded cluster architecture)
- Configure two-way data synchronization between ApsaraDB for MongoDB sharded cluster instances

#### Synchronize data from a ApsaraDB for MongoDB instance (replica set architecture) to another ApsaraDB for MongoDB instance (replica set architecture or sharded cluster architecture)

The following table describes the precautions and limits when you synchronize data to a MongoDB database, such as a self-managed MongoDB database or an ApsaraDB for MongoDB instance.

Category

Description

> Document Version: 20220712

Category	Description		
Limits on the source database	<ul> <li>Bandwidth requirements: The server to which the source database belongs must have a sufficient egress bandwidth. Otherwise, the data synchronization speed is affected.</li> <li>The collections to be synchronized must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select collections as the objects to synchronize and you want to edit collections (such as renaming collections), up to 1,000 collections can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 collections to synchronize, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements must be met: <ul> <li>The oplog feature must be enabled.</li> </ul> </li> <li>For an incremental data synchronization task, the oplogs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the oplogs of the source database must be stored for at least 7 days. After the full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, Data Transmission Service (DTS) may fail to obtain the oplogs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of oplogs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance</li> </ul>		
Other limits	<ul> <li>To ensure compatibility, the version of the destination MongoDB database must be the same as or later than the version of the source MongoDB database. If the version of the destination database is earlier than the version of the source database, database compatibility issues may occur.</li> <li>DTS cannot synchronize data from the admin or local database.</li> <li>Transaction information is not retained. When transactions are synchronized to the destination database, they are converted into a single record.</li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation is complete, the storage usage of collections in the destination database is larger than that of collections in the source database.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination database.</li> </ul>		

Category	Description	
Special cases	If the source database is a self-managed MongoDB database, take note of the following limits:	
	<ul> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> </ul>	
	• DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no update operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the latency of the synchronization task is too high, you can perform an update operation on the source database to update the latency.	
	<b>Note</b> If you select an entire database as the object to synchronize, you can create a heartbeat. The heartbeat is updated or receives data every second.	

#### Configure two-way data synchronization between ApsaraDB for MongoDB sharded cluster instances

The following table describes the precautions and limits when you synchronize data to a MongoDB database, such as a self-managed MongoDB database or an ApsaraDB for MongoDB instance.

Category

Category	Description		
Category Limits on the source and destination dat abases	<ul> <li>Description</li> <li>Bandwidth requirements: The server to which the source database belongs must have a sufficient egress bandwidth. Otherwise, the data synchronization speed is affected.</li> <li>The collections to be synchronized must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> <li>If you select collections as the objects to synchronize and you want to edit collections (such as renaming collections), up to 1,000 collections can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 collections to synchronize, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> <li>The following requirements must be met: <ul> <li>The oplog feature must be enabled.</li> <li>For an incremental data synchronization task, the oplogs of the source database must be stored for more than 24 hours. If you perform both full data</li> </ul> </li> </ul>		
	synchronization and incremental data synchronization, the oplogs of the source database must be stored for at least 7 days. After the full data synchronization is complete, you can set the retention period to more than 24 hours. Otherwise, Data Transmission Service (DTS) may fail to obtain the oplogs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of oplogs in accordance with the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not ensure service reliability and performance		
	• During a data synchronization task, ApsaraDB for MongoDB sharded cluster instances involved in the task cannot be scaled. Otherwise, the task fails.		

Category	Description		
	<ul> <li>To ensure compatibility, the version of the destination MongoDB database must be the same as or later than the version of the source MongoDB database. If the version of the destination database is earlier than the version of the source database, database compatibility issues may occur.</li> <li>If the source or the destination instance is located in a region outside the Chinese</li> </ul>		
	mainland, two-way data synchronization is supported only between instances located within the same region. For example, two-way data synchronization is supported between instances within the Japan (Tokyo) region. Two-way data synchronization between an instance in the Japan (Tokyo) region and another instance in the Germany (Frankfurt) region is not supported.		
	• DTS cannot synchronize data from the admin or local database.		
	• Transaction information is not retained. When transactions are synchronized to the destination database, they are converted into a single record.		
Other limits	• Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.		
	• During full data synchronization, concurrent INSERT operations cause fragmentation in the collections of the destination database. After the full data synchronization is complete, the storage usage of collections in the destination database is larger than that of collections in the source database.		
	• During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination databases. If you use tools other than DTS to write data to the destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.		
1			

## 3.10. Precautions and limits for synchronizing data from a Db2 for LUW database

This topic describes the precautions and limits that you must take note of when you synchronize data from a database. To ensure that your data synchronization task runs as expected, you must read the precautions and limits before you configure the task.

### Synchronize data from a Db2 for LUW database to a PolarDB-X 2.0 instance

**Note** By default, Data Transmission Service (DTS) disables FOREIGN KEY constraints for the destination database in a data synchronization task. Therefore, the cascade and delete operations of the source database are not synchronized to the destination databases.

Category	Description		
	<ul> <li>Bandwidth requirements: The server to which the source database belongs must have a sufficient egress bandwidth. Otherwise, the data synchronization speed is affected.</li> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> </ul>		
	• If you select tables as the objects to be synchronized and you want to edit tables (such as renaming tables or columns), up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables to be synchronized, configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.		
Limits on the	• The following requirements for data logs must be met:		
	<ul> <li>The data logging feature must be enabled. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>		
	<ul> <li>If you perform only incremental data synchronization, the data logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the data logs of the source database must be stored for at least seven days. After the full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the data logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of data logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability and performance.</li> </ul>		

Category	Description		
Other limits	<ul> <li>Schema synchronization is not supported. Before you configure a data synchronization task, you must create databases and tables in the destination instance.</li> <li>DTS synchronizes incremental data from a database to the destination database based on the Change Data Capture (CDC) replication technology of . However, the CDC replication technology has its own limits. For more information, see General data restrictions for SQL Replication.</li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.</li> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on objects. Otherwise, data synchronization may fail.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on the source tables. For more information, see Change schemas without locking tables.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. If you use tools other than DTS to write data to the destination database. How only out use DMS to perform online DDL operations.</li> </ul>		
Special cases	<ul> <li>You must take note of the following items because the source database is a selfmanaged database:</li> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the latency of the synchronization task is too high, you can perform a DML operation on the source database to update the latency.</li> <li>Note If you select an entire database as the object to be synchronized, you can create a heartbeat table. The heartbeat table is updated or receives data every second.</li> </ul>		

## Synchronize data from a DB2 for LUW database to an AnalyticDB for MySQL V3.0 cluster

**Note** By default, Data Transmission Service (DTS) disables FOREIGN KEY constraints for the destination database in a data synchronization task. Therefore, the cascade and delete operations of the source database are not synchronized to the destination databases.

Category	Description		
	<ul> <li>Bandwidth requirements: The server to which the source database belongs must have a sufficient egress bandwidth. Otherwise, the data synchronization speed is affected.</li> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> </ul>		
	<ul> <li>If you select tables as the objects to be synchronized and you need to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.</li> </ul>		
Limits on the	• The following requirements for data logs must be met:		
	<ul> <li>The data logging feature must be enabled. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>		
	<ul> <li>If you perform only incremental data synchronization, the data logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the data logs of the source database must be stored for at least seven days. After the full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the data logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of data logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability and performance.</li> </ul>		

Category	Description		
Other limits	<ul> <li>Schema synchronization is not supported. Before you configure a data synchronization task, you must create databases and tables in the destination instance.</li> <li>DTS synchronizes incremental data from a database to the destination database based on the Change Data Capture (CDC) replication technology of . However, the CDC replication technology has its own limits. For more information, see General data restrictions for SQL Replication.</li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination database. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may cause the loads of the database servers to increase.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the size of used tablespace of the destination database is larger than that of the source database.</li> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on objects. Otherwise, data synchronization may fail.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on the source tables. For more information, see Change schemas without locking tables.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database, by our write data to the destination database see on the source and destination database. This prevents data inconsistency between the source and the stination database. If you use tools other than DT to write data to the destination database. For more information, see Change schemas without locking tables.</li> <li>V3.0 has limits on the usage of disk space. If the disk space usage of the nodes in an V3.0 cluster reaches 80%, the DTS t</li></ul>		
Special cases	<ul> <li>You must take note of the following items because the source database is a selfmanaged database:</li> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the latency of the synchronization task is too high, you can perform a DML operation on the source database to update the latency.</li> <li>Note If you select an entire database as the object to be synchronized, you can create a heartbeat table. The heartbeat table is updated or receives data every second.</li> </ul>		

### Synchronize data from a Db2 for LUW database to a PolarDB for MySQL cluster

**Note** By default, DTS disables FOREIGN KEY constraints for the destination database in a data synchronization task. Therefore, the cascade and delete operations of the source database are not synchronized to the destination database.

Category	Description	
	<ul> <li>Bandwidth requirements: The server to which the source database belongs must have a sufficient egress bandwidth. Otherwise, the data synchronization speed is affected.</li> <li>The tables to be synchronized must have PRIMARY KEY or UNIQUE constraints, and all fields must be unique. Otherwise, the destination database may contain duplicate data records.</li> </ul>	
	• If you select tables as the objects to be synchronized and you need to edit tables (such as renaming tables or columns) in the destination database, up to 1,000 tables can be synchronized in a single data synchronization task. If you run a task to synchronize more than 1,000 tables, a request error occurs. In this case, we recommend that you split the tables and configure multiple tasks to synchronize the tables, or configure a task to synchronize the entire database.	
Limits on the source database	• The following requirements for data logs must be met:	
	<ul> <li>The data logging feature must be enabled. Otherwise, error messages are returned during precheck and the data synchronization task cannot be started.</li> </ul>	
	<ul> <li>If you perform only incremental data synchronization, the data logs of the source database must be stored for more than 24 hours. If you perform both full data synchronization and incremental data synchronization, the data logs of the source database must be stored for at least seven days. After the full data synchronization is completed, you can set the retention period to more than 24 hours. Otherwise, DTS may fail to obtain the data logs and the task may fail. In exceptional circumstances, data inconsistency or loss may occur. Make sure that you set the retention period of data logs based on the preceding requirements. Otherwise, the Service Level Agreement (SLA) of DTS does not guarantee service reliability and performance.</li> </ul>	

Category	Description		
Other limits	<ul> <li>Schema synchronization is not supported. Before you configure a data synchronization task, you must create databases and tables in the destination instance.</li> <li>DTS synchronizes incremental data from a database to the destination database based on the Change Data Capture (CDC) replication technology of . However, the CDC replication technology has its own limits. For more information, see General data restrictions for SQL Replication.</li> <li>Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours. During full data synchronization, DTS uses read and write resources of the source and destination databases. This may increase the loads of the database servers.</li> <li>During full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination database. After full data synchronization is completed, the tablespace of the destination database is larger than that of the source database.</li> <li>We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on objects. Otherwise, data synchronization may fail.</li> <li>If you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations on the source tables. For more information, see Change schemas without locking tables.</li> <li>During data synchronization, we recommend that you use only DTS to write data to the destination database. This prevents data inconsistency between the source and destination database, data loss may occur in the destination database when you use DMS to perform online DDL operations.</li> </ul>		
Special cases	<ul> <li>You must take note of the following items because the source database is a selfmanaged database:</li> <li>If you perform a primary/secondary switchover on the source database when the data synchronization task is running, the task fails.</li> <li>DTS calculates synchronization latency based on the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. If no DML operation is performed on the source database for a long time, the synchronization latency may be inaccurate. If the latency of the synchronization task is too high, you can perform a DML operation on the source database to update the latency.</li> <li>Note If you select an entire database as the object to be synchronized, you can create a heartbeat table. The heartbeat table is updated or receives data every second.</li> </ul>		

# 4.Prepare the database accounts for data synchronization

When you configure a data synchronization task, you must specify the database accounts for the source and destination databases. The database accounts are used for data synchronization. Different databases and synchronization types require different permissions. You must create and authorize database accounts before you configure a data synchronization task.

#### Permissions required for the source database account

Database	Required permission	References
ApsaraDB RDS for MySQL instance	Read permissions on the objects to synchronize	Create databases and accounts for an ApsaraDB RDS for MySQL instance and Modify the permissions of a standard account on an ApsaraDB RDS for MySQL instance
Self-managed MySQL database	The SELECT permission on the objects to synchronize The REPLICATION CLIENT, REPLICATION SLAVE, and SHOW VIEW permissions Permissions to create databases and tables. The permissions allow Data Transmission Service (DTS) to create a database named dts to record heartbeat data during synchronization.	Create an account for a user-created MySQL database and configure binary logging
PolarDB for MySQL cluster	Read permissions on the objects to synchronize	Create a database account
PolarDB for Oracle cluster	Permissions of a privileged account	Create database accounts
PolarDB-X 1.0 instance	Read permissions on the objects to synchronize	Manage database accounts
ApsaraDB RDS for SQL Server instance	The owner permission on the source database	Modify the permissions of a standard account on an ApsaraDB RDS for SQL Server instance
	<b>Note</b> A privileged account has the required permissions.	
Self-managed SQL Server database	The permissions of the sysadmin role	CREATE USER and GRANT (Transact-SQL)

Database	Required permission	References
ApsaraDB RDS for PostgreSQL instance	Permissions of a privileged account. The account must be the owner of the database.	
	<b>Note</b> If the source database runs on an ApsaraDB RDS for PostgreSQL instance V9.4 and you synchronize only DML operations, the database account must have the REPLICATION permission.	Create an account on an ApsaraDB RDS for PostgreSQL instance and Create a database on an ApsaraDB RDS for PostgreSQL instance
Self-managed PostgreSQL database	Permissions of the superuser role	CREATE USER and GRANT
ApsaraDB for Redis instance	Read permissions on the objects to synchronize	Create and manage database accounts
Self-managed Redis database	The <b>PSYNC</b> or <b>SYNC</b> statement can be executed on the source Redis database.	None
ApsaraDB for MongoDB instance	<ul> <li>Full data migration: read permissions on the source database</li> <li>Incremental data migration: read permissions on the source database, the admin database, and the local database</li> </ul>	For more information, see Manage user permissions on MongoDB databases.
Self-managed MongoDB database	<ul> <li>Full data migration: the read permissions on the source database</li> <li>Incremental data migration: the read permissions on the source database, the admin database, and the local database</li> </ul>	For more information, see db.createUser().
Self-managed TiDB database	The SELECT permission on objects to migrate and the SHOW VIEW permission	Privilege Management

#### Permissions required for the destination database account

Database	Required permission	References
ApsaraDB RDS for MySQL instance	Read and write permissions on the destination database	Create databases and accounts for an ApsaraDB RDS for MySQL instance and Modify the permissions of a standard account on an ApsaraDB RDS for MySQL instance
#### Data Synchronization Prepare the d atabase accounts for data synchro nization

Database	Required permission	References
Self-managed MySQL database	The ALL permission on the destination database	Create an account for a user-created MySQL database and configure binary logging
PolarDB for MySQL cluster	The ALL permission on the destination database	Create a database account
PolarDB for Oracle cluster	The permissions of the <b>database</b> owner	You can specify the <b>database owner</b> when you create a database.
PolarDB-X 1.0 instance	Write permissions on the objects to synchronize	Manage database accounts
ApsaraDB for	If you use the instance password, no authorization is required.	None
Redis instance	If you use a custom account, read and write permissions are required.	Create and manage database accounts
Self-managed Redis database	The database password must be valid.	None
ApsaraDB for MongoDB instance	The dbAdminAnyDatabase permission, read and write permissions on the destination database, and read permissions on the local database	For more information, see Manage user permissions on MongoDB databases.
Self-managed MongoDB database	The dbAdminAnyDatabase permission, the read and write permissions on the destination database, and the read permissions on the local database	For more information, see db.createUser().
AnalyticDB for MySQL cluster	<ul> <li>Version 2.0: DTS automatically creates a database account and grants permissions to the account. You do not need to specify the database account.</li> <li>Version 3.0: Read and write permissions are required.</li> </ul>	Version 3.0: Create a database account
AnalyticDB for PostgreSQL instance	The <b>initial account</b> or an account that has the RDS_SUPERUSER permission is required.	<ul> <li>Initial account: Create a database account</li> <li>An account that has the RDS_SUPERUSER permission: Manage users and permissions</li> </ul>

Database	Required permission	References
Message Queue for Apache Kafka instance	N/A         Image: The instance type of the Message Queue for Apache Kafka instance is VPC Instance, you do not need to specify the database account or database password.         None	
Self-managed Kafka cluster	N/A <b>Note</b> If no authentication is enabled for the Kafka cluster, you do not need to enter the username or password.	None
DataHub project	You do not need to specify the <b>database account</b> when you configure the task.	None
Elasticsearch cluster	The logon name and logon password that are specified when you create the Elasticsearch cluster. The default logon name is elastic.	Create an Elasticsearch cluster
MaxCompute project	The CREATE TABLE, CREATE INSTANCE, CREATE RESOURCE, CREATE JOB, and List permissions on the project to synchronize	When you configure the data synchronization task, DTS automatically authorizes the database account.
Tablestore	You do not need to specify the <b>database account</b> when you configure the task.	None
ClickHouse cluster	Read and write permissions on the objects to synchronize	Create an account

## Permissions required for the database accounts in two-way data synchronization tasks

The following table lists the permissions that are required for the source and destination database accounts in two-way data synchronization tasks. The permissions allow DTS to create a database named dts in the source and destination databases to prevent circular data replication.

Database	Required permission	References
ApsaraDB RDS for MySQL instance	Permissions of a privileged account	Create databases and accounts for an ApsaraDB RDS for MySQL instance
Self-managed MySQL database	The SELECT permission on the objects to synchronize The REPLICATION CLIENT, REPLICATION SLAVE, and SHOW VIEW permissions The permissions to create databases and tables. The permissions allow DTS to create a database named dts to prevent circular data replication.	Create an account for a user-created MySQL database and configure binary logging
ApsaraDB RDS for PostgreSQL instance	Permissions of a privileged account. The account must be the owner of the database.	Create an account on an ApsaraDB RDS for PostgreSQL instance <b>and</b> Create a database on an ApsaraDB RDS for PostgreSQL instance
Self-managed PostgreSQL database	Permissions of the superuser role	CREATE USER and GRANT
PolarDB for MySQL cluster	Permissions of a privileged account	Create a database account
ApsaraDB for	If you use the instance password, no authorization is required.	None
Redis instance	If you use a custom account, the read and write permissions are required.	Create and manage database accounts
Self-managed Redis database	The PSYNC or SYNC statement can be executed on the source Redis database.	None

## 5.Synchronization task management 5.1. Rename an object to be synchronized

By default, 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 Data Transmission Service (DTS) to rename the object in the destination instance.

#### Precautions

You can rename an object only in the **Select Objects to Synchronize** step when you configure a data synchronization task.

**Note** We recommend that you do not rename objects after the data synchronization task is started. Otherwise, the task may fail.

#### Procedure

1. In the Select Objects to Synchronize step, 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 an object, you can rename the object.

#### Data Synchronization Synchronizati

on task management

Processing Mode In					
Existed Target Table:   Pre-cheo	ck and Intercept $\bigcirc$ Ignore				
Available			Selected (To edit an object r Edit.) Learn more.	name or its filter, hover over	the object and cli
Expand the tree before you perform	a glol 🛛 🔍				
🕂 🚈 test db				Q	_
			dtstestdata (10bjects)		Edit
🗉 📂 test_db_rkwi_0001			i customer		
🗉 📂 test_db_rkwi_0002					
🕀 🚈 test_db_rkwi_0003		>			
🕀 🚈 test_db_rkwi_0004					
🕀 📂 test_db_rkwi_0005		<			
🕀 🚰 test_db_rkwi_0006					
🕀 🚰 test_db_rkwi_0007					
🖃 📑 dtstestdata					
🖃 📑 Tables					
Select All			Colort All		
*Rename Databases and Tables	Do Not Change Database a	ind Table Names	Change Database and Ta	ble Names	
Rename Databases and Tables.				bie names	
Retry Time for Failed Connection	720 Minutes	()			
*Target library object name case	● DTS default policy ○ Co	nsistent with source	e library		
policy:	<ul> <li>Consistent with the targe</li> </ul>	t library default pol	licy(Capitalize)		

- 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, enter the database name that you want to use in the destination instance.

ilso updated.	'ou edit the source database	name, the name of the d	estination database is
		Source Da	tabase Name:dtstestdat
* DatabaseName:	dtstestdatanew		Ì
DML and DDL Statement			
Filtering Se	elect DDL or DML stateme	nts. (j)	
DML Filter: [	🖌 insert 🔽 update 🔽 dele	te	
Table: 🔽	create 🔽 alter 🔽 drop 🔽	rename 🔽 truncate	
View: 🗸	create 🔽 alter 🔽 drop		
Procedure: 🔽	create 🔽 alter 🔽 drop		
Trig&Func: 🔽	create trigger 🔽 drop trigge	er 🔽 create function 🔽	drop function
Index: 🔽	create 🔽 drop		

#### • Table name mapping

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

Edit Table		$\times$
Information: After you corresponding table or o	u edit the table or column name in the source database, the column nam Source Table Name:customer	
* Table Name: cust	omernew	
Filter: state claus	supports the WHERE clause in SQL ements. Only data that meets the WHERE se can be migrated to the destination	
DML and DDL		
Filtering Sel	ect DDL or DML statements. (1)	
DML Filter: 🔽	insert 🗹 update 🔽 delete	
Table: 🗹 cr	eate 🔽 alter 🔽 drop 🔽 rename 🔽 truncate	
View: 🔽 cr	eate 🔽 alter 🔽 drop	
Procedure: 🗹 cr	eate 🔽 alter 🔽 drop	
Trig&Func: 🗹 cr	eate trigger $\checkmark$ drop trigger $\checkmark$ create function $\checkmark$ drop function	
Index: 🗹 cr	eate 🔽 drop	

#### • Column name mapping

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

✓ Select All	Column Name Source Column Na	ame:address
	addressnew	varchar(32)
<b>&gt;</b>	id	int(11)
<b>v</b>	name	varchar(32)
		ОК
⑦ Note	In this step, you can clear the columns that	at do not need to be synchronize

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

### 5.2. Use SQL conditions to filter data

When you select objects for a Data Transmission Service (DTS) task, you can specify SQL conditions to filter data. Only the data that meets the specified conditions is synchronized or migrated to the destination database. This feature is applicable to scenarios such as regular data synchronization or migration and table partitioning.

#### Prerequisites

The task is being configured. The current step is **Configure Objects and Advanced Settings**. For more information about how to create and configure a DTS task, see **Configure a data synchronization** task.

#### Limits

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

- 1. In the **Configure Objects and Advanced Settings** step, move a table to the **Selected Objects** section and then right-click the table.
- 2. In the Edit Table Name dialog box, enter one or more SQL conditions in the Filter Conditions field.

Edit <mark>T</mark> able Name		
Note: If you rename a tab	ole or column, the new name of the table or column takes effect in the destination database.	
Table Name:	order	
Filtering Conditions:	You can enter an SQL WHERE clause. Only the data that meets the WHERE clause is migrated to the destination database. Example: id>10	2
Select DDL and DML Operation	ns to Be Synchronized	
select DML Operations to Be Sj	delete	Select All
Select DDL Operations to Be Sy	nchronized	Select All
Table	🖌 drop ✔ rename ✔ truncate	
✔ View ✔ create ✔ alter	🖌 drop	
Procedure		
🗹 create 🗹 alter 🛛	🖌 drop	
Trig&Func create function	drop function 🗹 create trigger 🗹 drop trigger	
Index		
Column Name	Туре	

#### ? Note

- An SQL condition is a standard SQL WHERE statement. The following operators are supported:
   , != , < , > , and in . Only the data that meets the WHERE condition is synchronized or migrated to the destination database. In this example, enter order>100
- You can specify a time condition in an SQL WHERE statement. However, you must make sure that the specified time condition is valid. For example, to filter incremental data created after 2020, you must enter create\_time>'2020-01-01'
   Or create\_time>'2020'
   You cannot enter create time>'2020'
- You can use apostrophes (') in an SQL condition if necessary. For example, you can enter address in ('hangzhou', 'shanghai').
- Filter conditions are case-insensitive. If a table in the source database contains Column A and Column a, you can use an SQL WHERE statement to filter only Column A. In this case, you can enter WHERE A=10 rather than WHERE a=10.

#### 3. Click **OK**.

4. Configure other parameters that are required for the DTS task.

# 5.3. Specify the capitalization of object names in the destination instance

When you configure a Data Transmission Service (DTS) task, you can specify the capitalization of database names, table names, and column names in the destination instance.

#### Limits

When you specify the capitalization of object names in the destination instance, take note of the following limits:

- You cannot specify capitalization rules for views, functions, or stored procedures. If a table in the destination database is renamed, the corresponding views, functions, and stored procedures may fail to be created.
- The objects cannot contain CHECK constraints or computed columns.

#### **Capitalization rules**

DTS supports the following capitalization rules for object names:

• DTS default policy

DTS configures the capitalization of database names, table names, and column names in the destination instance based on the database type and related parameters, such as lower\_case\_tabl e names .

Destination database type	Destination database parameter	Capitalization of object names in the destination instance	
<ul> <li>Self-managed MySQL database or ApsaraDB RDS for MySQL instance</li> <li>ApsaraDB RDS for MariaDB TX</li> </ul>	The lower_case_table_names parameter is set to 1 or 2.	Database names and table names in the destination instance are in lowercase. The capitalization of column names in the destination instance is the same as that in the source instance.	
<ul> <li>PolarDB for MySQL</li> <li>PolarDB-X</li> <li>AnalyticDB for MySQL V3.0</li> </ul>	The lower_case_table_names parameter is set to 0.	The capitalization of database names, table names, and column names in the destination instance is the same as that in the source instance.	
<ul> <li>AnalyticDB for MySQL V2.0</li> <li>DataHub</li> <li>MaxCompute</li> <li>HybridDB for MySQL</li> </ul>	None	Database names, table names, and column names in the destination instance are in lowercase.	
Self-managed Oracle database	None	Database names, table names, and column names in the destination instance are in uppercase.	
<ul> <li>Self-managed SQL Server database or ApsaraDB RDS for SQL Server instance</li> <li>PolarDB O Edition</li> <li>Self-managed PostgreSQL database or ApsaraDB RDS for PostgreSQL instance</li> <li>ApsaraDB RDS for PPAS</li> <li>AnalyticDB for PostgreSQL</li> <li>Self-managed Db2 database</li> <li>Self-managed MongoDB database or ApsaraDB for MongoDB instance</li> <li>Self-managed Redis database or ApsaraDB for Redis instance</li> <li>Tablestore</li> <li>Elasticsearch</li> </ul>	None	Database names and table names in the destination instance are in lowercase. The capitalization of column names in the destination instance is the same as that in the source instance.	

• Consistent with the source database

The capitalization of database names, table names, and column names in the destination instance is the same as that in the source instance.

- Consistent with the default policy of the destination database (uppercase) The names of all the databases, tables, and columns that are migrated or synchronized to the destination instance are in uppercase.
- Consistent with the default policy of the destination database (lowercase) The names of all the databases, tables, and columns that are migrated or synchronized to the destination instance are in lowercase.

## 5.4. Modify the naming rules for additional columns

DTS adds additional columns to the tables that are synchronized to MaxCompute. These additional columns are used to manage metadata, sort data, and remove duplicates. If the names of additional columns are the same as the names of existing columns in the destination table, data synchronization fails. To avoid this issue, we recommend that you modify the naming rules for additional columns.

#### Context

Before you modify the naming rules for additional columns, check whether additional columns and existing columns in the destination table will have name conflicts. The following table lists the previous and new naming rules for additional columns.

Onte In step 2, you can specify whether to use the new naming rules for additional columns.

#### Naming rules for additional columns

Destination instance	Previous additional column name	New additional column name	Reference
MaxCompute	<ul> <li>record_id</li> <li>operation_flag</li> <li>utc_timestamp</li> <li>before_flag</li> <li>after_flag</li> <li>modifytime_year</li> <li>modifytime_mont</li></ul>	<ul> <li>new_dts_sync_record_id</li> <li>new_dts_sync_operation_flag</li> <li>new_dts_sync_utc_timestamp</li> <li>new_dts_sync_before_flag</li> <li>new_dts_sync_after_flag</li> <li>new_dts_sync_modifytime_year</li> <li>new_dts_sync_modifytime_mont</li></ul>	For more information, see
	h <li>modifytime_day</li> <li>modifytime_hour</li> <li>modifytime_minut</li>	h <li>new_dts_sync_modifytime_day</li> <li>new_dts_sync_modifytime_hour</li> <li>new_dts_sync_modifytime_minu</li>	Schema of an incremental
	e	te	data table.

- 1. Create and configure a data synchronization task. For more information, see steps 1 to 7 in Configure a data synchronization task (旧控制台).
- 2. In the Select Objects to Synchronize step, specify whether you want to use the new naming

#### rules for additional columns.

1.Select Source and Destination I	Instances for 2.Select (	Object to Be Syr	nchronized	3.P	Precheck	
Initial Synchronization:	Initial Schema Synchronization					
Initial Synchronization: Available Expand the tree before you	☑ Initial Schema Synchronization	> <	Selected (To edit an object name or its i Edit.) Learn more.	filter, hover ov	er the object and	click
Select All Change Mapped Name: "Whether to enable new addit column rules	<ul> <li>Do Not Change Database an</li> <li>ional</li> <li>Yes</li> <li>No</li> </ul>	d Table Names	Select All O Change Database and Table Names	Gangel	Proviner	Prochast
				Cancer	Previous	Precheck
Option	Description					
Yes	If you select <b>Yes</b> , the ne <b>Note</b> If the pre names of existing colu conflicts.	w naming vious name umns in the	rules for additional colu es of additional column e destination table, sele	imns are s are the ct <b>Yes</b> to	used. e same as t o avoid na	the Ime
	DTS adds the <pre>new_dts_sync_ prefix to the previous names of additional columns. For example, if the previous name of an additional column is <pre>record_id</pre>, the new name of the additional column is <pre>new_dts_sync_record_id</pre>.</pre>			the		
No	If you select No, the pre-	vious nami	ng rules for additional c	olumns	are used.	

**Note** For more information about the naming rules for additional columns, see Naming rules for additional columns.

3. Configure other parameters that are required for the data synchronization task.

#### **Related topics**

- Overview of data synchronization scenarios
- Overview of data synchronization scenarios

## 5.5. View the connection status and performance of data synchronization

In the Data Transmission Service (DTS) console, you can view the performance of full data synchronization and incremental data synchronization. DTS provides the connection and performance metrics to help you manage data synchronization tasks.

- 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 data synchronization instance resides.
- 4. On the Synchronization Tasks page, click the ID of the data synchronization instance.
- 5. Perform one of the following operations:
  - View the connection status and performance of full data synchronization
    - a. In the left-side navigation pane, choose **Performance Monitoring > Performance of Full Data Synchronization**.
    - b. On the page that appears, the connection status and performance of full data synchronization are displayed. You can select a time range to view the trend charts of performance metrics for full data synchronization.

全量同步链路拓技	r C				调整全量迁移速率
		BPS: 0.00 MB/s RPS: 0.00 Row/s	C	BPS: 0.00 MB/s RPS: 0.00 Row/s	Eliza
土里内罗注能					
选择时间范围:	2020年4月21日 10:39	- 2020年4月21日 11:39	<b>m</b>		
	全量同步	地流量(Mb/s)	指标会义	源、目标实例RPS	<del>新版会义</del> 严
Section		Description			

Section	Description
Topology of Full Data Synchronization	<ul> <li>In this section, you can view the read/write performance and network information about the connections between DTS and the source and destination databases. The following parameters are provided:</li> <li>Connection between DTS and the source database</li> <li>BPS: the amount of data that DTS reads from the source database per second. Unit: MB/s.</li> <li>RPS: the number of records that DTS reads from the source database per second.</li> <li>Network Latency: the network latency between DTS and the source database.</li> <li>Connection between DTS and the destination database</li> <li>BPS: the amount of data that DTS writes to the destination database per second. Unit: MB/s.</li> <li>RPS: the number of records that DTS writes to the destination database per second. Unit: MB/s.</li> <li>RPS: the number of records that DTS writes to the destination database per second. Unit: MB/s.</li> <li>RPS: the number of records that DTS writes to the destination database per second. Unit: MB/s.</li> <li>RPS: the number of records that DTS writes to the destination database per second.</li> <li>Network Latency: the network latency between DTS and the destination database per second.</li> </ul>
	In this section, you can view the bandwidth, records per second (RPS), read/write response time, and network latency.
Performance of Full Data Synchronization	<b>Note</b> To view the description of performance metrics, move the pointer over the <b>Indicator Meaning</b> button at the upper-right corner of a trend chart.

- View the connection status and performance of incremental data synchronization
  - a. In the left-side navigation pane, choose **Performance Monitoring > Performance of Increment al Data Synchronization**.
  - b. On the page that appears, the connection status and performance of incremental data synchronization are displayed. You can select a time range to view the trend charts of performance metrics for incremental data synchronization.



Section	Description
Topology of Incremental Data Synchronization	<ul> <li>In this section, you can view the data transmission status and network latency between DTS modules and the source and destination databases. DTS modules include the data collection module, the data cache module, and the data writing module. The following parameters are provided:</li> <li>BPS: the bandwidth between DTS modules. Unit: MB/s.</li> <li>RPS: the number of records that are transmitted between DTS modules per second.</li> </ul>
	<ul> <li>Network Latency: the network latency between DTS modules and the source and destination databases.</li> </ul>

Section	Description
	In this section, you can view performance information such as the bandwidth, synchronization speed, and synchronization latency. You can select metrics from the <b>More Metrics</b> drop-down list. The following metrics are provided:
	<ul> <li>Bandwidth (MB/s): the bandwidth of data that the data writing module pulls from the data pulling module per second.</li> </ul>
Performance of Incremental Data Synchronization	<ul> <li>Synchronization Speed (TPS): the number of transactions that DTS synchronizes to the destination database per second.</li> </ul>
	<ul> <li>Synchronization Latency: the difference between the timestamp of the latest synchronized data in the destination database and the current timestamp in the source database. Unit: milliseconds.</li> </ul>
	<ul> <li>Number of DDL Operations: the number of data definition language (DDL) operations that are executed in the destination database within the selected time range.</li> </ul>
	Latency (ms): the difference between the time when a data record is generated in the source database and the time when the data record is written to the destination database. For example, if a data record is generated in the source database at 7 o'clock and DTS writes the data record to the destination database at 8 o'clock, the task is delayed by 1 hour.
	<ul> <li>Slow SQL Queries: the number of slow SQL queries that are generated in the destination database within the selected time range.</li> </ul>

## 5.6. Reset a data synchronization task

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

#### Prerequisites

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

#### Impacts on billing

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

- 1. Log on to the DTS console.
- 2. In the left-side navigation pane, click **Data Synchronization**.
- 3. In the upper part of the **Synchronization Tasks** page, select the region where the data synchronization instance resides.
- 4. Find the data synchronization task, and choose More > Reset Task in the Actions column.

• Warning If you reset a data synchronization task, the following impacts take place. Proceed with caution.

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

#### References

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

## 5.7. Add an object to a data synchronization task

When a data synchronization task is running, you can add objects to the task or remove objects from the task. You do not need to configure the task again. This topic describes how to add an object to a data synchronization task.

#### Prerequisites

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

#### Precautions

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

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

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscriptic Upgrade
Pause Task Delete Task			Total: 1 item(s), P	2 Modify Ol Ver Page: 20 Configure	bjects to Synchronize
				Reset Tas	sk
				Delete Ta	isk

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.

<page-header></page-header>	ionfigure Source and Destination 2.Select Objects to Synchronize	3.Advanced Settings > 4.Precheck	
Available     If you search globally, please expand the:     Image: Tables     Image: Tables <th>Synchronization Mode:One-Way Synchronization (DML+DDL) Proccessing Mode In Existed Target Table:</th> <th></th> <th></th>	Synchronization Mode:One-Way Synchronization (DML+DDL) Proccessing Mode In Existed Target Table:		
Select All Select All Select All	Available If you search globally, please expand the Q S S S S S S S S S S S S S	Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.	
*Name batch change:   No  Yes	Select All	Select All	
	*Name batch change:   No  Yes		

#### 6. Click Precheck.

#### ? Note

- After you reselect the objects to be synchronized, DTS performs a precheck on the data synchronization task. The data synchronization task can be started only after it passes the precheck.
- If the task fails to pass the precheck, click the next to each failed item to view

details. Troubleshoot the issues based on the causes and run a precheck again.

7. Start the data synchronization task.

## 5.8. Remove an object from a data synchronization task

When a data synchronization task is running, you can add objects to the task or remove objects from the task. You do not need to configure the task again. This topic describes how to remove an object from a data synchronization task.

#### Prerequisites

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

#### Precautions

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

#### Procedure

- 1. Log on to the DTS console.
- 2. In the left-side navigation pane, click Data Synchronization.
- 3. At the top of the **Synchronization Tasks** page, select the region where the data synchronization 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 **i**con to move the object to the **Available** section.

Synchronization Mode:One-Way Synchronization (DM Proccessing Mode In Existed Target Table:  Pre-check and Interce Available If you search globally, please expand the   C Sys distestdata Tables	AL+DDL) ept Ignore	Selected (To edit an object name e Edit.) Learn more.	or its filter, hover over the	e object and click	
Select All		Select All			
*Name batch change:      No      Yes					

#### 6. Click Precheck.

#### ? Note

- After you reselect the objects to be synchronized, DTS performs a precheck on the data synchronization task. The data synchronization task can be started only after it passes the precheck.
- If the task fails to pass the precheck, click the icon next to each failed item to view

details. Troubleshoot the issues based on the causes and run a precheck again.

7. Start the data synchronization task.

### 5.9. 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.

• If a data synchronization task is stopped, the task cannot be restarted. To synchronize data again, you must reconfigure the task.

#### Procedure

- 1. Log on to the DTS console.
- 2. At the top of the **Synchronization Tasks** page, select the region where the data synchronization instance resides.
- 3. On the **Synchronization Tasks** page, click the ID of the data synchronization instance.
- 4. Find the data synchronization task, and choose More > Stop Task in the Actions column.



#### References

Release pay-as-you-go instances

### 5.10. Troubleshoot precheck failures

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

Precheck item	Cause of failure	Solution
	The database account or password is invalid.	Obtain the valid password, and change the password of the source or destination database in the DTS console. The password is specified when you configure the data synchronization task. For more information, see Change the password of a database account.

विश्वसंहर्णनाही connectivity	Cause of failure	Solution
Destination database connectivity	The IP address that is used to access the source or destination database is disallowed to access the database.	<ul> <li>For a MySQL database, authorize the database account again. The CIDR blocks of DTS servers must be added to the IP whitelist. For more information about the CIDR blocks of DTS servers in each region, see Add the CIDR blocks of DTS servers to the security settings of on-premises databases.</li> <li>For an SQL Server database, disable the firewall or trigger settings.</li> <li>For an Oracle database, set the TC P.VALIDNODE_CHECKING parameter to no and restart the process.</li> </ul>
	A firewall is configured on the server where the source or destination database resides.	Disable the firewall settings.
	The network between DTS servers and the source or destination database is unavailable.	Contact Alibaba Cloud engineers for technical support. For more information, visit submit a ticket .
Source database version	DTS does not support the current database version. For more information, see Overview of data synchronization scenarios.	Upgrade or downgrade your database version.
Database existence	<ul> <li>If the destination database does not exist in the destination instance, DTS may fail to create a database because of the following reasons:</li> <li>The database name contains special characters except letters, digits, underscores (_), and hyphens (-).</li> <li>The character set of the database is not UTF8, GBK, Latin1, or UTF8MB4.</li> <li>The account that is used to access the destination database does not have the required permissions.</li> </ul>	Create a database in the destination instance or grant the required permissions to the database account.

Precheck item	Cause of failure	Solution
Source database permissions	The account that is used to access the source or destination database does not have the required permissions.	
Destination database permissions	<b>Note</b> The permissions that are required for a database vary with the database type. For more information, see <b>Overview of data synchronization scenarios</b> .	Grant the required permissions to the database account.
Schema name conflict	The source and destination databases have the same name, or the source and destination tables have the same name.	<ul> <li>Log on to the destination database, and change the database name or table names.</li> <li>Map the names of the conflicting objects to other objects in the destination database by using the object name mapping feature. For more information, see Rename an object to be synchronized.</li> <li>Remove the conflicting objects. For more information, see Remove an object from a data synchronization task.</li> </ul>
Value of server_id in the source database	The value of the server-id parameter is not set to an integer that is greater than or equal to 2.	Log on to the source database and change the value of the server-id parameter. For more information, see Value of server_id in a source database.
Whether binary logging is enabled for the source database	The binary logging feature is disabled for the source database.	Log on to the source database and enable the binary logging feature. For more information, see Source database binlogging.
Binary log format of the source database	The binary log format of the source database is set to ROW.	Log on to the source database, run the set global binlog_format='ROW'; command, and then restart the MySQL process.

Precheck item	Cause of failure	Solution
Integrity of FOREIGN KEY constraint	The parent table on which a child table depends is not included in the required objects. This impairs the integrity of the FOREIGN KEY constraint.	<ul> <li>Remove the foreign key dependencies from the child table that fails to pass the precheck.</li> <li>Add the parent table to the required objects. For more information, see Add an object to a data synchronization task.</li> <li>Remove the child table from the required objects. For more information, see Remove an object from a data synchronization task.</li> </ul>
Storage engine	The storage engine of the source table is FEDERATED, MRG_MyISAM, or TokuDB.	Log on to the source database and set the storage engine of the source table to InnoDB.
Character set	DTS does not support the character set of the required objects, such as UCS2.	Log on to the source database and set the character set of source tables to UTF8, GBK, Latin1, or UTF-8MB4.
Complicated topologies	DTS does not support the topology that you use for the source and destination instances. For more information, see Synchronization topologies.	Stop the conflicting task or wait for the task to complete, and then perform data synchronization again.
Format of MySQL database password	The format of the password that is used to access the source database is no longer valid.	Change the format of the database password. For more information, visit old_passwords.

## 5.11. Data formats of a Kafka cluster

When you use Data Transmission Service (DTS) to migrate or synchronize data to a Kafka cluster, you can select the format in which data records are stored. This topic describes the definition of the data formats to facilitate your data parsing.

#### Data formats

DTS allows you to store data into a Kafka cluster in the following formats:

- DTS Avro: A data serialization format into which data structures or objects can be converted.
- Shareplex JSON: The format in which the data read from the source database by using the data replication software SharePlex is stored.
- Canal JSON: The format in which data is stored in a Kafka cluster after Canal parses the logs about the incremental data of the source database and transmits the incremental data to a Kafka cluster.

#### DTS Avro

**DTS Avro** is the default data format. Data migrated or synchronized to a Kafka cluster by using DTS is stored in the Avro format. You must parse the data based on the schema definition of DTS Avro. For more information, visit GitHub.

#### Shareplex JSON

The following table describes the parameters related to the Shareplex JSON format.

Parameter	Description
time	The UTC time when the transaction in the database is committed. It is in the yyyy-MM-ddTHH:mm:ssZ format.
userid	The ID of the user who commits the transaction.
op	The operation type. Valid values: INSERT, UPDATE, DELETE, TRUNCATE, DROP COLUMN, UPDATE BEFORE, and UPDATE AFTER.
scn	The system change number (SCN) that identifies the version of the transaction that the database commits at a specific time. Each committed transaction is assigned a unique SCN.
rowid	A relatively unique address value that is used to identify a record in the database.
trans	The ID of the transaction.
seq	The sequence number of the operation in the transaction. It starts from 1.
size	The total number of operations in the transaction.
table	The table name.
idx	The index of the operation in the transaction. It is in the seq/size format. For example, 1/11 indicates that the sequence number of the operation is 1 in the transaction that contains a total number of 11 operations.
posttime	The time when the transaction is committed to the destination database.

#### Examples

• Data inserted

#### Dat a Transmission Service

```
{
   "meta": {
       "time": "2017-06-16T14:24:34",
       "userid": 84,
       "op": "ins",
         "scn": "14589063118712",
         "rowid": "AAATGpAAIAAItcIAAA",
       "trans": "7.0.411499",
       "seq": 1,
       "size": 11,
       "table": "CL BIZ1.MIO LOG",
        "idx": "1/11",
       "posttime": "2017-06-16T14:33:52"
   },
   "data": {
      "MIO_LOG_ID": "32539737"
    }
}
```

#### • Data updated

```
{
   "meta": {
       "time": "2017-06-16T15:38:13",
       "userid": 84,
       "op": "upd",
       "table": "CL BIZ1.MIO LOG"
        . . .
   },
   "data": {
      "CNTR NO": "1171201606"
   },
    "key": {
       "MIO_LOG_ID": "32537893",
       "PLNMIO REC ID": "31557806",
       "POL_CODE": null,
       "CNTR TYPE": null,
       "CNTR_NO": "1171201606syui26"
   }
}
```

#### • Data deleted

```
{
    "meta": {
        "time": "2017-06-16T15:51:35",
        "userid": 84,
        "op": "del",
     },
    "data": {
        "MIO_LOG_ID": "32539739",
        "PLNMIO_REC_ID": "31557806",
        "POL_CODE": null,
        "CNTR_TYPE": null,
        "CG_NO": null
     }
}
```

#### Canal JSON

The following table describes the parameters related to the Canal JSON format.

Parameter	Description
database	The database name.
	The time when the operation is performed on the database. The value is a 13-bit UNIX timestamp. Unit: milliseconds.
es	<b>Note</b> You can use a search engine to obtain a UNIX timestamp converter.
id	The serial number of the operation.
isDdl	<ul> <li>Indicates whether the operation is a DDL operation.</li> <li>true: The operation is a DDL operation.</li> <li>false: The operation is not a DDL operation.</li> </ul>
mysqlType	The data type of the field.
	The data before update.
old <b>and</b> data	<b>Note</b> For change tracking instances that are created before March 20, 2022, the value of old is data and the value of data is NULL. To keep consistent with the open source community, the data value is data and the old value is NULL for change tracking instances that are created or restarted as of March 20, 2022.
pkNames	The name of the primary key.
sql	The SQL statement.

Parameter	Description		
sqlType	The data type of the field after conversion. For example, the type LONG is converted from UNSIGNED INTEGER and BIGDECIMAL from UNSIGNED LONG.		
table	The table name.		
ts	The time when the data is written to the destination database. The value is in the 13-bit UNIX timestamp format. Unit: milliseconds.		
	<b>Note</b> You can use a search engine to obtain a UNIX timestamp converter.		
type	The operation type. Valid values: DELETE, UPDATE, and INSERT.		

#### Examples of data updated

**Note** For change tracking instances that are created before March 20, 2022, after the DELETE statements of the source table is synchronized, the old value is data and the data value is NULL. To keep consistent with the open source community, the data value is data and the old value is NULL for change tracking instances that are created or restarted as of March 20, 2022.

#### Change tracking instances that are created before March 20, 2022

```
{
   "old": [
      {
           "shipping type": "aaa"
       }
   ],
   "database": "dbname",
   "es": 1600161894000,
   "id": 58,
   "isDdl": false,
   "mysqlType": {
       "id": "bigint(20)",
       "shipping_type": "varchar(50)"
   },
   "pkNames": [
       "id"
   ],
   "sql": "",
   "sqlType": {
       "id": -5,
       "shipping_type": 12
   },
   "table": "tablename",
   "ts": 1600161894771,
   "type": "DELETE"
}
```

#### Change tracking instances that are created or restarted as of March 20, 2022

```
{
    "data": [
       {
           "id": "500000287",
           "shipping type": null
       }
    ],
    "database": "dbname",
    "es": 1600161894000,
    "id": 58,
    "isDdl": false,
    "mysqlType": {
       "id": "bigint(20)",
       "shipping_type": "varchar(50)"
    },
    "pkNames": [
       "id"
    ],
    "sql": "",
    "sqlType": {
       "id": -5,
       "shipping type": 12
    },
    "table": "tablename",
    "ts": 1600161894771,
    "type": "DELETE"
}
```

#### Example of a DDL operation

```
{
    "database": "dbname", The name of the source database.
    "es": 1600161894000, The time when the data is written to the binary logs of the sou
rce database.
    "id": 58, The offsets of DTS cache.
    "isDdl": true, Indicates that the operation is a DDL operation.
    "sql": "eg: create xxx", The DDL statements recorded in the binary logs.
    "table": "tablename", The name of the table that is synchronized.
    "ts": 1600161894771, The time when the data is written to the destination database
by using DTS.
    "type": "DDL"
}
```

## 5.12. Specify the policy for synchronizing data to Kafka partitions

When you configure a task to synchronize data to a Kafka cluster, you can specify the policy for synchronizing data to Kafka partitions. The policy allows you to improve the synchronization performance. For example, you can synchronize data to different partitions based on hash values.

#### Hash algorithm

Data Transmission Service (DTS) uses the hashCode() method in Java to calculate hash values.

#### Configuration method

In the **Select Objects to Synchronize** step of a task creating wizard, you can specify the policy for synchronizing data to Kafka partitions. For more information, see Synchronize data from an ApsaraDB RDS for MySQL instance to a self-managed Kafka cluster and Overview of data synchronization scenarios.

• Warning After a data synchronization task is started, do not change the number of partitions in the destination topic. Otherwise, data synchronization fails.

Svnc	hron	ization	policies
Jync	mon	ization	policies

Policy	Description	Advantage and disadvantage	
Synchronize All Data to Partition 0	DTS synchronizes all data and DDL statements to Partition 0 of the destination topic.	<ul> <li>Advantage: The order in which all objects are created and changed is the same as that in the source database.</li> <li>Disadvantage: This policy provides ordinary synchronization performance.</li> </ul>	
	DTS uses the database and table names as the partition key to calculate the hash value. Then, DTS synchronizes the data and DDL statements of each table to the corresponding partition of the destination topic.	<ul> <li>Advantage: The order in which a destination table is created and changed is the same as that of the source table.</li> </ul>	
Synchronize Data to Separate Partitions Based on Hash Values of Database and Table Names	<ul> <li>Note</li> <li>The data and DDL statements of the same table are synchronized to the same partition.</li> <li>If a DDL statement is irrelevant to a table, for example, CREATE DAT ABASE, the statement is synchronized to Partition 0.</li> </ul>	<ul> <li>This policy provides good synchronization performance.</li> <li>Disadvantage: Tables are synchronized to different partitions. After data synchronization, the order of data changes on different tables may become inconsistent.</li> </ul>	

Policy	Description	Advantage and disadvantage
Synchronize Data to Separate Partitions Based on Hash Values of Primary Keys	DTS uses a table column as the partition key to calculate the hash value. The table column is the primary key by default. If a table does not have a primary key, the unique key is used as the partition key. DTS synchronizes each row to the corresponding partition of the destination topic. You can specify one or more columns as partition keys to calculate the hash value.           ⑦         Note           •         If you use this policy, DDL statements are synchronized to Partition 0 of the destination topic by default.           •         If a table does not have a primary key or unique key, DTS synchronizes the data and DDL statements of the table to Partition 0 of the destination topic.	<ul> <li>Advantage: This policy provides the best synchronization performance.</li> <li>Disadvantage: After data synchronization, the order of data changes on each data record remains the same. However, the order of data changes on different tables or tables without a primary key may become inconsistent.</li> </ul>

## 6.Synchronize data between MySQL databases

### 6.1. Configure two-way data synchronization between MySQL instances

Data Transmission Service (DTS) supports two-way data synchronization between two MySQL databases. This feature is applicable to scenarios such as active geo-redundancy (unit-based) and geodisaster recovery. This topic describes how to configure two-way data synchronization between ApsaraDB RDS for MySQL instances. You can also follow the procedure to configure data synchronization tasks for self-managed MySQL databases.

#### Prerequisites

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

#### Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization 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 you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations during data synchronization. For more information, see Change schemas without locking tables.

Q Warning If you use tools other than DTS to write data to the destination database, we recommend that you do not use DMS to perform online DDL operations. Otherwise, data loss may occur in the destination database.

If the source or destination instance of a two-way data synchronization task resides in a region outside the Chinese mainland, data can be synchronized only within this region. Cross-region two-way synchronization is not supported. For example, if the source instance resides in the Japan (Tokyo) region, data can be synchronized only within the Japan (Tokyo) region and cannot be synchronized to the Germany (Frankfurt) region.

#### Supported synchronization topologies

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



#### Supported databases

The following table lists the types of 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 MySQL databases.

Source database	Destination database	
<ul> <li>ApsaraDB RDS for MySQL instance</li> <li>Self-managed database that is hosted on Elastic</li></ul>	<ul> <li>ApsaraDB RDS for MySQL instance</li> <li>Self-managed database that is hosted on Elastic</li></ul>	
Compute Service (ECS) <li>Self-managed database that is connected over</li>	Compute Service (ECS) <li>Self-managed database that is connected over</li>	
Express Connect, VPN Gateway, or Smart Access	Express Connect, VPN Gateway, or Smart Access	
Gateway	Gateway	
<ul> <li>Self-managed database that is connected over</li></ul>	<ul> <li>Self-managed database that is connected over</li></ul>	
Database Gateway	Database Gateway	
• Self-managed database that is connected over	• Self-managed database that is connected over	
Cloud Enterprise Network (CEN)	Cloud Enterprise Network (CEN)	

#### SQL operations that can be synchronized

#### **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 specify 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 specify.

#### ♥ 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 cannot guarantee that the conflict detection mechanism can prevent all data conflicts. To perform two-way synchronization, make sure that records with the same primary key, business primary key, or unique key are updated only on one of the synchronization nodes.
- DTS provides conflict resolution policies to prevent conflicts that may occur during data synchronization. You can select a conflict resolution policy when you configure two-way data synchronization.

#### Limits

• Incompatibility with triggers

If you select a database as the object to be synchronized and the database contains a trigger that updates a table, data inconsistency may occur. 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 prevent 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 a data synchronization task for a source database that contains a trigger.

- Limits on RENAME TABLE operations RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if you select a table as the object and rename the table during data synchronization, the data of this table is not synchronized to the destination database. To prevent 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 two-way data synchronization, you can synchronize DDL operations only in one direction. If DDL synchronization in a direction is configured, DDL synchronization in the opposite direction is not supported. Only DML operations can be synchronized in the opposite direction.

#### Procedure

1. Purchase an instance for two-way data synchronization. For more information, see Purchase a data synchronization instance.

Notice On the buy page, set both Source Instance and Destination 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 set parameters for each task.

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) -	Actions
			Pay-As-You-Go	Two-Way Synchronization	Switch to Subscription Upgrade View Synchronization Task ~ More
Task Name	Status	Synchronization Details	Source/Destination Instance		Actions
August August Select	Not Configured		Not Configured Not Configured	Co	nfigure Synchronization Channel
	Not Configured		Not Configured Not Configured	Co	nfigure Synchronization Channel

6. Configure source and destination instances.

1.Configure Source and Destination	2.Select Objects to Synchroniz	re 🔪 🗄	3.Advanced Settings	$\rangle$	4.Precheck
Synchronization Task Name:	RDS		]		
			-		
Source Instance Details					
Technology Transv					
instance type:	RDS Instance	•			
Instance Region:	Singapore				
* Instance ID:	rm-	•	RDS Instances of Other Apsara	Stack Accounts	
* Database Account:	dtstest		]		
* Database Password:	•••••	\$			
Destination Instance Details					
Instance Type:	PDC Instance				
Instance Posien:	Cingapore	KUS Instance			
instance Region.	Singapore				
* Instance ID:	rm-				
* Database Account:	dtstest				
* Database Password:	•••••	]			
				Cancel	Set Whitelist and Next
Section	Parameter	Description			
		DTS autom	atically generates a	a task nam	e. We
N/A	Synchronization Task	recommend	that you specify a	an informat	ive name to
	Name	identify the	task. You do not r	need to use	e a unique task
		name.			

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

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

1.Select Source and Destination > 2.Select Object to Be Synchronized	3.Advanced Settings 🔰 4.Precheck
Synchronization Mode:Two-Way Synchronization (DML+DDL) Exclude DDL Statements: Yes  No DML Statements for Synchronization: Insert I Delete Update	
Conflict Resolution Policy: TaskFailed (When a conflict occurs, an error is reported and the	e task 🗸
Available  Expand the tree before you perform a glo Q  forrecycle_bin  s asd for chw02  for dts for dtstest0512_jzhz_0001_ext_0001  for dtstest123  for dtstestdata1  for sys	Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.
Select All	Select All
<ul> <li>*Rename Databases and Tables:</li> <li>Do Not Change Database and Table Names</li> <li>*Source table DMS_ONLINE_Do you want to copy the temporary table to the target database during DDL:</li> </ul>	Change Database and Table Names
Retry Time for Failed Connection     720     Minutes     ⑦	Cancel Previous Next

Setting	Parameter	Description
	Exclude DDL Statements	<ul> <li>To exclude DDL operations, select Yes.</li> <li>To include DDL operations, select No.</li> <li>Notice Limits on DDL synchronization direction: To ensure the stability of two-way data synchronization, you can synchronize DDL operations only in the forward direction.</li> </ul>
	DML Statements for Synchronizatio n	Select the types of DML operations that you want to synchronize. By default, the <b>INSERT</b> , <b>UPDATE</b> , and <b>DELETE</b> operations are selected. You can select the DML operation types based on your business requirements.
	Conflict Resolution Policy	<ul> <li>Select the resolution policy for synchronization conflicts. By default, TaskFailed is selected. You can select a conflict resolution policy based on your business requirements.</li> <li>TaskFailed <ul> <li>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.</li> <li>Ignore <ul> <li>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.</li> </ul> </li> <li>Overwrite <ul> <li>If a conflict occurs during data synchronization, the conflicting records in the destination database are used.</li> </ul> </li> </ul></li></ul>

Seteintgthe	Parameter	Description
n policy		<ul> <li>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.</li> <li>Notice You can use the object name mapping feature to rename the tables that are synchronized to the destination database. You can use this feature if the source and destination databases contain identical table names and the tables in the destination database cannot be deleted or renamed. For more information, see Rename an object to be synchronized.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> </ul>
Select proces mode conflic tables	Select the processing mode of conflicting tables	<ul> <li>Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks.</li> <li>During initial data synchronization, DTS does not synchronize the data records that have the same primary keys as the data records in the destination database. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization.</li> <li>If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only specific columns are synchronized or the data synchronization task fails.</li> </ul>

Setting	Parameter	Description
		Select one or more objects (tables or a database) from the <b>Available</b> section and click the <b>&gt;</b> icon to move the objects to the <b>Selected</b> section.
Select the objects to be synchronized	N/A	<ul> <li>Notice</li> <li>If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination instance, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Rename an object to be synchronized.</li> </ul>
Rename Databases and Tables	N/A	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.
	N/A	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> </ul>
Replicate Temporary Tables When DMS Performs DDL Operations		<b>Note</b> If online DDL operations generate a large amount of data, the data synchronization task may be delayed.
		• <b>No</b> : DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.
		<b>Note</b> If you select No, the tables in the destination database may be locked.

Setting	Parameter	Description		
Retry Time for Failed Connections		By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.		
	N/A	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.		

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

1.Configure Source and Destination	$\geq$	2.Select Objects to	Synchronize		3.Advanced Settings		4.Precheck	
Initial Synchronization:	Initial S	chema Synchronization	Initial Full Data	Synchronization				
						2 mil	Duriture Court	Product
						Cancel	Previous Save	Precheck

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 part of the page, click **Next: Precheck and Start Task**.

#### ♥ 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, you can click the next to each failed item

to view details.

- You can trouble shoot the issues based on the causes and run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed 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 the task by following Steps 5 to 12.

			Pay-As-You-Go	Switch Two-Way Synchronization View	to Subscription   Upgrade v Synchronization Task ~   More
Task Name	Status	Synchronization Details	Source/Destination Instance		Actions
singapore-singapore-medium	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	rm- rm-		Pause Task More
singapore-singapore-medium	Not Configured		rm- rm-	Configure Synchron	ization Channel

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

			Pay-As-You-Go	Two-Way Synchronization	Switch to Subscription   Upgrade View Synchronization Task~   More
Task Name	Status	Synchronization Details	Source/Destination Instance		Actions
singapore-singapore-medium	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	rm- rm-		Pause Task   More
singapore-singapore-medium	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	rm-		Pause Task More

# 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 instances are created. For more information, see Create an ApsaraDB RDS for MySQL instance.
- The database type of the source and destination ApsaraDB RDS instances is MySQL.

#### Precautions

• DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization 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 you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations during data synchronization. For more information, see Change schemas without locking tables.

Q Warning If you use tools other than DTS to write data to the destination database, we recommend that you do not use DMS to perform online DDL operations. Otherwise, data loss may occur in the destination database.

- The tables to be migrated in the source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After initial full data synchronization is complete, the size of used 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

## Limits

• Incompatibility with triggers

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

Limits on RENAME TABLE operations

RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A is selected as the object to synchronize and is renamed Table B, Table B cannot be synchronized to the destination database. To prevent this situation, you can select the entire database where Table A is located as the object to synchronize when you configure the data synchronization task.

## Procedure

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

**?** Note Set both Source Instance and Destination Instance to 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. In the upper part of the **Synchronization Tasks** page, select the region where the data synchronization instance resides.
- 5. Find the data synchronization instance and click **Configure Task** in the Actions column.
- 6. Configure the source and destination instances.

1.Configure Source and De	stination	2.Select Objects to Synchronize	>	3.Advanced Settings	$\rangle$	4.Precheck
Synchronization Task	Name: RDS MySOL					
	10011/042					
Source Instance Details						
Instanc	e Type: RDS Instance	9	•			
Instance	Region: China (Hangzh	ou)				
* Insta	nce ID: rm-bp		•	RDS Instances of Other Apsara S	tack Accounts	
* Database A	ccount: dtstest					
* Database Pa	ssword:		<b>\$</b> >			
* Enci	ryption: 🖲 Non-encryp	ed SSL-encrypted				
Destination Instance Details						
Instanc	e Type: RDS Instance	3	v			
Instance	Region: China (Hangzh	ou)				
* Insta	nce ID: rm-bp		•			
* Database A	ccount: dtstest					
* Database Pa	ssword:	•	<b>4</b> >			
* Enci	ryption:   Non-encryption:	ed OSSL-encrypted				
					Cancel	Set Whitelist and Next
Section	Parameter	Description	ו			
N/A	Synchroniz n Task Na	the task n that you sp the task. Y	ame that I pecify a de ou do not	DTS automatically scriptive name tha need to use a unic	generates. We It makes it eas que task name	e recommend sy to identify e.
	Instance T	ype Select <b>RDS</b>	Instance			
	Instance Region	The source this param	e region th eter canno	at you selected on ot be changed.	the buy page	e. The value of
	Instance II	D The ID of t	he source	RDS instance.		

Section	Parameter	Description				
		The database account of the source RDS instance.				
Source Instance Details	Database Account	<b>Note</b> If the database engine of the source RDS instance is <b>MySQL 5.5</b> or <b>MySQL 5.6</b> , you do not need to configure the <b>database account</b> or <b>database password</b> .				
	Database Password	The password of the database account.				
	Encryption	Select <b>Non-encrypted</b> or <b>SSL-encrypted</b> . If you want to select <b>SSL-encrypted</b> , you must enable SSL encryption for the ApsaraDB RDS instance before you configure the data synchronization task. For more information, see Configure SSL encryption for an ApsaraDB RDS for MySQL instance.				
		<b>Notice</b> The <b>Encryption</b> parameter is available only for regions in the Chinese mainland and the China (Hong Kong) region.				
	Instance Type	Select RDS Instance.				
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.				
	Instance ID	The ID of the destination ApsaraDB RDS instance.				
		The database account of the destination ApsaraDB RDS instance.				
Destination	Dat abase Account	<b>Note</b> If the database engine of the destination RDS instance is <b>MySQL 5.5</b> or <b>MySQL 5.6</b> , you do not need to configure the <b>database account</b> or <b>database password</b> .				
Instance Details	Dat abase Password	The password of the database account.				
	Encryption	Select <b>Non-encrypted</b> or <b>SSL-encrypted</b> . If you want to select <b>SSL-encrypted</b> , you must enable SSL encryption for the ApsaraDB RDS instance before you configure the data synchronization task. For more information, see Configure SSL encryption for an ApsaraDB RDS for MySQL instance.				
	Encryption	<b>Notice</b> The <b>Encryption</b> parameter is available only for regions in the Chinese mainland and the China (Hong Kong) region.				

7.

8. Select the synchronization policy and the objects to synchronize.

1.Select Source and Destination	on 2.Select Object to Be Synchronized	3.Advanced Settings > 4.Pr	recheck
Synchronization Mode:O	ne-Way Synchronization (DML+DDL)		
Available Expand the tree before you the precycle_bin the precycle_bi	u perform a glol Q 101_ext_0001	Selected (To edit an object name or its filter, hover over the Edit.) Learn more.	object and click
Select All		Select All	
<ul> <li>Rename Databases and Tab</li> <li>Source table DMS_ ONLINE, want to copy the temporary to the target database during DI</li> <li>Retry Time for Failed Connert</li> </ul>	les:      Do Not Change Database and Table I     Do you     Yes     No     O     Yes     No     O     to     to     T20     Minutes     O	Names O Change Database and Table Names	
		Cancel P	revious Next
Setting	Description		

Setting	Description
	<ul> <li>Select one or more objects from the Available section and click the &gt; icon to</li> <li>add the objects to the Selected section.</li> <li>You can select tables or databases as the objects to synchronize.</li> <li>Note <ul> <li>If you select a database as the object to synchronize, all schema changes in the database are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination database the name of the object remains unchanged. You can use</li> </ul> </li> </ul>
Select the objects to be synchronized	the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Rename an object to be synchronized.
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.
Replicate	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> </ul>
Replicate Temporary Tables When DMS Performs DDL Operations	<ul> <li>Note If online DDL operations generate a large amount of data, the data synchronization task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> <li>Note If you select No, the tables in the destination database may</li> </ul>
	be locked.

Setting	Description			
Retry Time for Failed Connections	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.			
	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.			

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

#### 10. Select the initial synchronization types.

	2.Select Objects to	Synchronize	3.Advanced Settings		
Initial Synchronization: 🔽 I	nitial Schema Synchronization	✓ Initial Full Data Synchroniza	tion		
			Can	cel Previous	Save Precheck

#### ? Note

- During initial synchronization, DTS synchronizes the schemas and data of 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. In most cases, you need to select both Initial Schema Synchronization and Initial Full Data Synchronization.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the 🕡 icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.
- 12. Close the **Precheck** dialog box after the following message is displayed: **Precheck Passed**. Then, the data synchronization task starts.

13. Wait until initial synchronization is complete and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscription Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < <b>1</b> > »

# 6.3. Synchronize data from a selfmanaged MySQL database hosted on ECS to an ApsaraDB RDS for MySQL instance

This topic describes how to synchronize data from a self-managed MySQL database hosted on Elastic Compute Service (ECS) to an ApsaraDB RDS for MySQL instance by using Data Transmission Service (DTS).

#### Prerequisites

- The version of the self-managed 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 ApsaraDB RDS for MySQL instance.

## Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization 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 you use only DTS to write data to the destination database, you can use Data Management (DMS) to perform online DDL operations during data synchronization. For more information, see Change schemas without locking tables.

Q Warning If you use tools other than DTS to write data to the destination database, we recommend that you do not use DMS to perform online DDL operations. Otherwise, data loss may occur in the destination database.

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

• 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.

## 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

Onte For more information about two-way synchronization, see Configure two-way data synchronization between MySQL instances.

# SQL operations that can be synchronized

## Limits

• Incompatibility with triggers

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

• Limits on RENAME TABLE operations

RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A is selected as the object to synchronize and is renamed Table B, Table B cannot be synchronized to the destination database. To prevent this situation, you can select the entire database where Table A is located as the object to synchronize when you configure the data synchronization task.

## Before you begin

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

#### Procedure

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

Once 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.

dtsl8dd3rk116tob07 RD5	Not Configured	Pay-As-You-Go	One-Way Synchronization	Configure Task Switch to Subscription Upgrade More

6. Configure the source and destination instances.

1.Configure Source and Destination	n 2.Select Objects to Synchronize	>	3.Advanced Settings	>	4.Precheck
Synchronization Tack Namo	-in				
Synchronization rask wante.	singapore-singapore-medium				
Source Instance Details					
Taskan en Turner					
Instance Type:	User-Created Database in ECS Instance	*			
Instance Region:	Singapore				
* ECS Instance ID:	THE DESIGNATION OF THE PARTY OF	-			
Database Type:	MySQL				
* Port Number:	3306				
* Database Account:	dtstest				
* Database Password:	•••••	4>			
Destination Instance Details					
Instance Type:	RDS Instance	•			
Instance Region:	Singapore				
* Instance ID:	rm-	•			
* Database Account:	dtstest				
* Database Password:	•••••	<b>\$</b> >			
				Cancol	Cat Whitelist and Next

Section	Parameter	Description
N/A	Synchronizat ion Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
	lnstance Type	Select User-Created Database in ECS Instance.
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.
	ECS Instance ID	Select the ID of the ECS instance that hosts the self-managed MySQL database.
	Database Type	The value of this parameter is set to <b>MySQL</b> and cannot be changed.
Source Instance Details	Port Number	Enter the service port number of the self-managed MySQL database. The default port number is <b>3306</b> .

Section	Parameter	Description		
	Database Account	Enter the account of the self-managed MySQL database. The account must have the SELECT permission on the required objects, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.		
	Database Password	Enter the password for the account of the self-managed MySQL database.		
	lnstance Type	Select RDS Instance.		
Instance Region Instance ID		The destination region that you selected on the buy page. You cannot change the value of this parameter.		
		Select the ID of the destination RDS instance.		
Destination Instance	Database Account	Enter the database account of the destination RDS instance. <b>Note</b> 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 of the destination database account.		
	Encryption	Select <b>Non-encrypted</b> or <b>SSL-encrypted</b> . If you want to select <b>SSL-encrypted</b> , 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 ApsaraDB RDS for MySQL instance.		
		<b>Notice</b> The <b>Encryption</b> parameter is available only for regions in the Chinese mainland and the China (Hong Kong) region.		

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

8. Select the synchronization policy and the objects to be synchronized.

at a bet ween MySQL dat abases

	none may synamonization (Brier BBE)				
Available			Selected (To edit an object name or Edit.) Learn more.	its filter, hover over	the object and click
Expand the tree before	you perform a glol 🔰 🔍			Q	
			📔 dtstestdata		
🕀 🚘 dts	0001 ext 0001	×			
		>			
🕀 📴 dtstestdata1 🕀 🚰 sys		<			
Calact All			Select All		
Select All					
Select All *Rename Databases and T	ables: O Not Change Datab	case and Table Names	Change Database and Table Name	nes	
Select All *Rename Databases and T *Source table DMS_ ONLIN want to copy the temporar the target database during	ables:      Do Not Change Datab IE_Do you     Yes     No	base and Table Names	Change Database and Table Nan	nes	
Select All *Rename Databases and T *Source table DMS_ ONLIN want to copy the temporar the target database during * Retry Time for Failed Con	ables:  Do Not Change Datables:  NE_Do you Yes No O L:  nnection	utes	Change Database and Table Nan	nes	

Setting	Description
	Select one or more objects from the <b>Available</b> section and click the > icon to add the objects to the <b>Selected</b> section. You can select tables or databases as the objects to synchronize.
Select the objects to be synchronized	<ul> <li>Note</li> <li>If you select a database as the object to synchronize, all schema changes in the database are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Rename an object to be synchronized.</li> </ul>
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.
Replicate	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> <li>Yote If online DDL operations generate a large amount of data,</li> </ul>
Temporary Tables When DMS Performs DDL Operations	<ul> <li>the data synchronization task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> </ul>
	Onte If you select No, the tables in the destination database may be locked.

Setting	Description		
Retry Time for	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.		
Retry Time for Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.		

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

10. Configure initial synchronization.

1.Configure Source and Destination	2.Select Objects to	Synchronize	3.Advanced Settings	4.Precheck
Initial Synchronization: 🗹 Init	tial Schema Synchronization	✓ Initial Full Data Synchronization		
			Cance	Previous Save Precheck

- 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.
   In most cases, you need to select both Initial Schema Synchronization and Initial Full Data Synchronization.
- 11. In the lower part of the page, click Next: Precheck and Start Task.

#### ♥ 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, you can click the icon next to each failed item

to view details.

- You can troubleshoot the issues based on the causes and run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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.



# 6.4. Synchronize data from a selfmanaged 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 self-managed 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 ApsaraDB RDS for MySQL instance.
- The version of the self-managed MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The self-managed MySQL database is connected to Alibaba Cloud VPC over Express Connect, VPN Gateway, or Smart Access Gateway. For more information, see Connect an on-premises database to DTS by using CEN.

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

## Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization 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.
- The tables to be migrated in the source database must have PRIMARY KEY or UNIQUE constraints and

all fields must be unique. Otherwise, the destination database may contain duplicate data records.

• 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

#### Limits

- Incompatibility with triggers
   If you select a database as the object to synchronize and the database contains a trigger that
   updates a table, data inconsistency may occur. For more information about how to solve this issue,
   see Configure a data synchronization task for a source database that contains a trigger.
- Limits on RENAME TABLE operations
   RENAME TABLE operations may cause data inconsistency between the source and destination
   databases. For example, if only Table A is selected as the object to synchronize and is renamed Table
   B, Table B cannot be synchronized to the destination database. To prevent this situation, you can
   select the entire database where Table A is located as the object to synchronize when you configure
   the data synchronization task.

## Before you begin

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

## Procedure

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

Once 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.

dtsl8dd3rk116tob07 RDS	Not Configured	Pay-As-You-Go	One-Way Synchronization	Configure Task Switch to Subscription Upgrade More

6. Configure the source and destination instances.

1 Configure Source and D	octination	2.0	Coloct Objects to Synchroniza	<u> </u>	2 Advanced Cottings		4 Drochock
1.conligure Source and D	Resultation	2.3	select objects to synchronize		5.Advanced Settings		4.PTECHECK
Synchronization Ta	ask Name:	RDS					
Source Instance Details							
Instance Type: User-Created			abase Connected Over Express Conne	ct, VPN Gatev 🔻			
Instanc	e Region:	Singapore					
* Peer VPC: vnr				•			
Datab	ase Type:	MySQL					
* IP	Address:	172.16.					
* Port	t Number:	3306					
* Database	Account:	dtstest					
* Database F	Password:	•••••		4>			
Destination Instance Details							
Insta	nce Type:	RDS Instance		•			
Instanc	e Region:	Singapore					
* Ins	stance ID:	rm-					
* Database	Account:	dtstest					
* Database (	Dogword	utstest					
* Database Password:		•••••		47			
							Cancel Set Whitelist and Next
Section	Para	meter	Description				
Synchronizat N/A ion Task Name		DTS automaticall specify an inform use a unique task	ly generat native nam k name.	es a task name. W ne for easy identifi	'e recon cation. `	וmend that you You do not need to	
Inst Typ		nce	Select User-Created Dat VPN Gateway, or Smart		abase Connecte Access Gateway	d over y.	Express Connect,
Inst Reg Pee		nce on	The source region that you selected on the buy page. You cannot change the value of this parameter.			e. You cannot	
		VPC	Select the ID of t database.	he VPC th	at is connected to	the self	f-managed MySQL

The value of this parameter is set to MySQL and cannot be changed.

Instance Details Database

Section	Parameter	Description
	Port Number	Enter the service port number of the self-managed MySQL database. The default port number is <b>3306</b> .
	Database Account	Enter the account of the self-managed MySQL database. The account must have the SELECT permission on the required objects, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.
	Database Password	Enter the password of the source database account.
	Instance Type	Select RDS Instance.
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.
	Redis Instance ID	Select the ID of the destination RDS instance.
Destination Instance Details	Database Account	Enter the database account of the destination RDS instance. <b>Note</b> 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 of the destination database account.
	Encryption	Select Non-encrypted or SSL-encrypted. If you want to select SSL- encrypted, you must enable SSL encryption for the RDS instance before you configure the data synchronization task. For more information, see Configure SSL encryption for an ApsaraDB RDS for MySQL instance.

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

## ? Note

8. Select the synchronization policy and the objects to be synchronized.

1.Select Source and Destination	on 2.Select Object to Be Synchronized	3.	Advanced Settings	>	4.Precheck
Synchronization Mode: <b>O</b>	ne-Way Synchronization (DML+DDL)				
Available Expand the tree before you Expand the tree bef	u perform a glol Q 01_ext_0001	Sele Edit	ected (To edit an object name of) Learn more.	or its filter, hover ove	r the object and click
Select All		Sala	oct All		
*Rename Databases and Tabl *Source table DMS_ONLINE, want to copy the temporary ta the target database during DD * Retry Time for Failed Connergiant	es:      Do Not Change Database and Table Do you     Yes     No     Yes     No     Zo     Minutes	e Names (	Change Database and Table Na	ames	
				Cancel	Previous Next
Setting	Description				

Setting	Description				
Select the objects to be synchronized	<ul> <li>Select one or more objects from the Available section and click the &gt; icon to</li> <li>add the objects to the Selected section.</li> <li>You can select tables or databases as the objects to synchronize.</li> <li>Note <ul> <li>If you select a database as the object to synchronize, all schema changes in the database are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination, see Rename an object to be synchronized.</li> </ul> </li> </ul>				
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.				
Replicate Temporary Tables When DMS Performs DDL Operations	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> <li>Note If online DDL operations generate a large amount of data, the data synchronization task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> <li>Note If you select No, the tables in the destination database may be locked.</li> </ul>				

Setting	Description			
Retry Time for	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.			
Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.			

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

10. Configure initial synchronization.

1.Configure Source and Destination	>	2.Select Objects to	Synchronize		3.Advanced Settings		4.Precheck	
Initial Synchronization: 🔽	Initial Sch	ema Synchronization	Initial Full Data	Synchronization				
						Cancel	Previous Save	Precheck

- During an 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.
   In most cases, you need to select both Initial Schema Synchronization and Initial Full Data Synchronization.
- 11. In the lower part of the page, click Next: Precheck and Start Task.

#### ♥ 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, you can click the icon next to each failed item

to view details.

- You can troubleshoot the issues based on the causes and run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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.



# 6.5. Synchronize data from an ApsaraDB RDS for MySQL instance to a self-managed 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 selfmanaged MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway by using Data Transmission Service (DTS).

## Prerequisites

• The version number of the self-managed 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 self-managed MySQL database is connected to a virtual private cloud (VPC) over Express Connect, VPN Gateway, or Smart Access Gateway. For more information, see Connect an on-premises database to Alibaba Cloud.

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

#### Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After initial full data synchronization is complete, the size of used

tablespace of the destination instance is larger than that of the source instance.

#### Limits

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

## 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

## Limits

• Incompatibility with triggers

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

Limits on RENAME TABLE operations

RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A is selected as the object to synchronize and is renamed Table B, Table B cannot be synchronized to the destination database. To prevent this situation, you can select the entire database where Table A is located as the object to synchronize when you configure the data synchronization task.

## Procedure

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

Onte Select MySQL for both the source and destination instances. 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. In the upper part of the **Synchronization Tasks** page, select the region where the data synchronization instance resides.
- 5. Find the data synchronization instance and click **Configure Task** in the Actions column.
- 6. Configure the source and destination instances.

#### Data Synchronization Synchronize d

at a bet ween MySQL dat abases

Synchronization Task Name:	MySQL						
Source Instance Details							
Instance Type:	RDS Instance						
Instance Region:	China (Hangzhou)						
* Instance ID:	rm-bp 🗸	RDS Instances of Other Apsara Stack Accounts					
* Database Account:	dtstest						
* Database Password:	······ Ø>						
* Encryption:	Non-encrypted      SSL-encrypted						
Destination Instance Details							
Instance Type:	User-Created Database Connected Over Express Connect, VPN Gatev 🔻						
Instance Region:	China (Hangzhou)						
* Peer VPC:	vpc-bp						
Database Type:	MySQL						
* IP Address:	172.16						
* Port Number:	3306						
* Database Account:	dtstest						
* Database Password:	••••••						

Section Parameter Description Synchronizat The task name that DTS automatically generates. We recommend that N/A ion Task you specify a descriptive name that makes it easy to identify the task. Name You do not need to use a unique task name. Instance Select RDS Instance. Туре Instance The source region that you selected on the buy page. The value of this Region parameter cannot be changed. Instance ID The ID of the ApsaraDB RDS for MySQL instance. The database account of the source ApsaraDB RDS instance. Database **Note** If the database type of the source ApsaraDB RDS Account instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password. Database Source The password of the database account. Password Instance Details

Section	Parameter	Description			
	Encryption	Select Non-encrypted or SSL-encrypted. If you want to select SSL- encrypted, you must enable SSL encryption for the ApsaraDB RDS instance before you configure the data synchronization task. For more information, see Configure SSL encryption for an ApsaraDB RDS for MySQL instance.			
	lnstance Type	Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway.			
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.			
	Peer VPC	The ID of the VPC that is connected to the self-managed database.			
	Database Type	This parameter is set to <b>MySQL</b> and cannot be changed.			
Destination	IP address	The server IP address of the self-managed MySQL database.			
Instance Details	Port Number	The service port number of the self-managed MySQL database. Default value: <b>3306</b> .			
		The account of the self-managed MySQL database.			
	Database Account	<b>Note</b> The account must have the SELECT permission on the objects to synchronize and the REPLICATION CLIENT, REPLICATION SLAVE, and SHOW VIEW permissions.			
	Database Password	The password of the database account.			

7.

8. Select the synchronization policy and the objects to synchronize.

		stable and a subject name of its mean noter over the object and
Expand the tree before you perform a provide the tree be	giol Q	□ Q
Select All		Celect All
	Do Not Change Database and Table Names	Change Database and Table Names
*Rename Databases and Tables:		

Setting	Description
Select the objects to be synchronized	<ul> <li>Select one or more objects from the Available section and click the ) icon to add the objects to the Selected section.</li> <li>You can select tables or databases as the objects to synchronize.</li> <li>Note <ul> <li>If you select a database as the object to synchronize, all schema changes in the database are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination, see Rename an object to be synchronized.</li> </ul> </li> </ul>
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.

Setting	Description			
	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> </ul>			
Replicate Temporary Tables When DMS	<b>Note</b> If online DDL operations generate a large amount of data, the data synchronization task may be delayed.			
Performs DDL Operations	• <b>No</b> : DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.			
	<b>Note</b> If you select No, the tables in the destination database may be locked.			
Retry Time for	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.			
Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.			

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

#### 10. Select the initial synchronization types.

1.Configure Source and Destination	>	2.Select Objects to	Synchronize		3.Advanced Settings		4.Precheck	
Initial Synchronization: 🗹 Initial Schema Synchronization 🛛 🗹 Initial Full Data Synchronization								
						Cancel	Previous Save	Precheck

#### ? Note

- During initial synchronization, DTS synchronizes the schemas and data of 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. In most cases, you need to select both Initial Schema Synchronization and Initial Full Data Synchronization.
- 11. In the lower-right corner of the page, click **Precheck**.

? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.
- 12. Close the **Precheck** dialog box after the following message is displayed: **Precheck Passed**. Then, the data synchronization task starts.
- 13. Wait until initial synchronization is complete and the data synchronization task enters the **Synchronizing** state.

You can view the state of the data synchronization task on the Synchronization Tasks 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 are created. For more information, see Create an ApsaraDB RDS for MySQL instance.
- The database type of the source and destination ApsaraDB RDS instances is MySQL.
- The source and destination ApsaraDB RDS for MySQL instances have internal endpoints.

#### Precautions

• DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.

- •
- The tables to be migrated in the source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After initial full data synchronization is complete, the size of used 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

#### Limits

• Incompatibility with triggers

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

• Limits on RENAME TABLE operations

RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if only Table A is selected as the object to synchronize and is renamed Table B, Table B cannot be synchronized to the destination database. To prevent this situation, you can select the entire database where Table A is located as the object to synchronize when you configure the data synchronization task.

## Preparations

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

**?** Note To authorize the Alibaba Cloud account that owns the destination instance, you must log on to the Resource Access Management (RAM) console with the Alibaba Cloud account that owns the source instance. Then, you can create a data migration 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 ApsaraDB RDS instance. For more information, see Purchase a data synchronization

instance.

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

- 2. Use the Alibaba Cloud account that owns the destination RDS instance to 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 Task** in the Actions column.

	dtsl8dd3rk116tob07 RDS	Not Configured	Pay-As-You-Go	One-Way Synchronization	Configure Task Switch to Subscription Upgrade More
--	---------------------------	----------------	---------------	----------------------------	---

6. Configure the source and destination instances.

1.Configure Source and D	estinatior	2.Select (	Dbjects to Synchronize	$\rangle$	3.Advanced Settings	>	4.Precheck	
Synchronization Task Name: RDS								
Source Instance Details	Source Instance Details							
Insta	nce Type:	RDS Instance		T				
Instanc	e Region:	Singapore						
*Apsara Stack Tenant Account 1	D of RDS							
	Instance				Guide			
*Ro	le Name:	ram-for-dts			Authorize Role Across Accounts			
* RDS Ins	RDS Instance ID:			•	RDS Instances of Current Account			
Destination Instance Details	Destination Instance Details							
Insta	nce Type:	RDS Instance		•				
Instanc	Instance Region: Since		Singapore					
* Instance ID:								
insurce is.								
* Database Account:		dtstest						
* Database Password:		هې						
						Cancel	Set Whitelist and Next	
Section	Parameter		Description					
N/A	Synchronizatio n Task Name		The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.			e recommend sy to identify e.		
	Instance Type		Select RDS Instance.					

Section	Parameter	Description					
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.					
Source Instance Details	Alibaba Cloud Account ID of RDS Instance	The ID of the Alibaba Cloud account that owns the source ApsaraDB RDS instance. Note Before you configure this parameter, click RDS Instances of Other Alibaba Cloud Accounts in the Source Instance Details section. Instance Type: RDS Instance Instance Region: Singapore Instance ID: RDS Instances of Other Apsara Stack Accounts					
	Role Name	The name of the RAM role that you configured earlier in Preparations.					
	RDS Instance ID	The ID of the source instance.					
	Instance Type	Select RDS Instance.					
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.					
	Instance ID	The ID of the destination instance.					
Destination Instance Details	Dat abase Account	The database account of the destination instance.  Note If the database engine of the destination instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password parameter.					
	Dat abase Password	The password of the database account.					
	Encryption	Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the ApsaraDB RDS instance before you configure the data synchronization task. For more information, see Configure SSL encryption for an ApsaraDB RDS for MySQL instance.					

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

8. Select the synchronization policy and the objects to synchronize.

1.Select Source and Destination 2.Select Object to Be Synchronized	3.Advanced Settings > 4.Precheck
Synchronization Mode:One-Way Synchronization (DML+DDL)          Available         Expand the tree before you perform a glol         Image: perform a glol	Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.
Select All	Select All
<ul> <li>Rename Databases and Tables:</li> <li>Do Not Change Database and Table Names</li> <li>Source table DMS_ ONLINE_ Do you want to copy the temporary table to the target database during DDL:</li> <li>Retry Time for Failed Connection</li> <li>Minutes ?</li> </ul>	Change Database and Table Names
	Cancel Previous Next
Setting Description	

Setting	Description			
Select the objects to be synchronized	<ul> <li>Select one or more objects from the Available section and click the &gt; icon to add the objects to the Selected section.</li> <li>You can select tables or databases as the objects to synchronize.</li> <li>Note <ul> <li>If you select a database as the object to synchronize, all schema changes in the database are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination database, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Rename an object to be synchronized.</li> </ul> </li> </ul>			
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.			
Replicate Temporary Tables When DMS Performs DDL Operations	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online operations.</li> <li>Note If online DDL operations generate a large amount of data, the data synchronization task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> <li>Note If you select No, the tables in the destination database may be locked.</li> </ul>			

Setting	Description
Retry Time for	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

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

10. Select the initial synchronization types.

1.Configure Source and Destination	$\geq$	2.Select Objects to	Synchronize		3.Advanced Settings		4.Precheck	
Initial Synchronization: 🔽	Initial Sci	hema Synchronization	Initial Full Data	a Synchronization				
						Cancel	Draviaura Cava	Drochock
						Cancel	Previous Save	Precheck

- During initial synchronization, DTS synchronizes the schemas and data of 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.
   In most cases, you need to select both Initial Schema Synchronization and Initial Full Data Synchronization.
- 11. In the lower part of the page, click Next: Precheck and Start Task.

#### ♥ 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, you can click the icon next to each failed item

to view details.

- You can troubleshoot the issues based on the causes and run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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.

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) <del>-</del>	Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscription Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < <u>1</u> > »

## 7.Synchronize data from a MySQL database to a different type of database

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

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

## Prerequisites

A PolarDB for MySQL cluster is created. For more information, see Create a 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 loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- •
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source database.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination cluster may contain duplicate data records.

## SQL operations that can be synchronized

## 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, see Synchronization topologies.

## Limits

- Incompatibility with triggers If you select a database as the object to be synchronized and the database contains a trigger that updates a table, data inconsistency may occur. For more information about how to solve this issue, see Configure a data synchronization task for a source database that contains a trigger.
- Limits on RENAME TABLE operations RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if you select a table as the object and rename the table during data synchronization, the data of this table is not synchronized to the destination database. To prevent this situation, you can select the database to which this table belongs as the object when you configure the data synchronization task.

## Procedure

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

**?** Note On the buy page, set Source Instance to MySQL, set Destination 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.

1.Configure Source and Destination In	stances 2.Select Objects to Synchronize	$\geq$	3.Advanced Settings	A.Precheck
Conductive Tech Name				
Synchronization Task Name:	RDS_TO_PolarDB			
Source Instance Details				
Instance Type:	RDS Instance	Ŧ		
Instance Region:	Singapore			
* Instance ID:	rm-gs!	•	RDS Instances of Other Apsara Stack Account	5
* Database Account:	dtstest			
* Database Password:	•••••	<b>\$</b> >		
Destination Instance Details				
Instance Type:	PolarDB			
Instance Region:	Singapore			
* PolarDB Instance ID:	pc-gs	-		
* Database Account:	dtstest			
* Database Password:	•••••	<b>\$</b> >		
				Cancel Set Whitelist and Next

Section	Parameter	Description		
None	Synchroniz at ion T ask Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.		
	lnstance Type	Select RDS Instance.		
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.		
		The database account of the source ApsaraDB RDS for MySQL instance.		
Source Instance Details	Dat abase Account	<ul> <li>Note</li> <li>The account must have the SELECT permission on the objects to synchronize and the REPLICATION CLIENT, REPLICATION SLAVE, and SHOW VIEW permissions.</li> <li>If the source ApsaraDB RDS for MySQL instance runs MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password.</li> </ul>		
	Database Password	The password of the database account.		
	Encryption	Select <b>Non-encrypted</b> or <b>SSL-encrypted</b> . If you want to select <b>SSL-encrypted</b> , you must enable SSL encryption for the ApsaraDB RDS for MySQL instance before you configure the data synchronization task. For more information, see Configure SSL encryption for an ApsaraDB RDS for MySQL instance.		
		<b>Notice</b> The <b>Encryption</b> parameter is available only for regions in the Chinese mainland and the China (Hong Kong) region.		
	lnstance Type	This parameter is set to <b>PolarDB</b> and cannot be changed.		
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.		
	PolarDB Instance ID	The ID of the destination PolarDB for MySQL cluster.		
Destination		The database account of the source PolarDB cluster.		
Instance Details	Database Account	<b>Note</b> The database account must have the ALL permission on the objects to synchronize.		

Section	Parameter	Description
	Database Password	The password of the database account.

7.

8. Select the processing mode of conflicting tables and the objects to be synchronized.

1.Select Source and Destination 2.Select Object to Be Synchronized	3.Advanced Settings A.Precheck
Synchronization Mode: One-Way Synchronization (DML+DDL)	
Available  Expand the tree before you perform a glol Q  forrecycle_bin_ for asd for chw02 for dts for dtstest0512_jzhz_0001_ext_0001 for dtstest123 for dtstestdata1 for sys	Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.
Select All	Select All
<ul> <li>*Rename Databases and Tables:</li> <li>*Source table DMS_ONLINE_Do you want to copy the temporary table to the target database during DDL:</li> <li>* Retry Time for Failed Connection</li> </ul>	Change Database and Table Names
	Cancel Previous Next
Setting Description	

Setting	Description
Select the processing mode of conflicting tables	<ul> <li>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.</li> <li>Note You can use the object name mapping feature to rename the tables that are synchronized to the destination database. You can use this feature if the source and destination databases contain identical table names and the tables in the destination database cannot be deleted or renamed. For more information, see Rename an object to be synchronized.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>During initial data synchronization, DTS does not synchronize the data records that have the same primary keys as the data records in the destination database. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization.</li> <li>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.</li> </ul>
Select the objects to be synchronized	<ul> <li>Select one or more objects from the Available section and click the &gt; icon to move the objects to the Selected section. You can select tables or databases as the objects to be synchronized.</li> <li>Note <ul> <li>If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination cluster, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination, see Rename an object to be synchronized.</li> </ul> </li> </ul>

Setting	Description		
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.		
Replicate Temporary Tables When DMS Performs DDL Operations	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> <li>Note If online DDL operations generate a large amount of data, the data synchronization task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> <li>Note If you select No, the tables in the destination database may be locked.</li> </ul>		
Retry Time for Failed Connections	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails. <b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.		

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

#### 10. Select the initial synchronization types.

L.Configure Source and Destination $>$	<ul> <li>2.Select Objects to</li> </ul>	Synchronize	3.Advanced Settings		4.Precheck
Initial Synchronization: 🔽 In	itial Schema Synchronization	Initial Full Data Synchron	ization		
				Cancel Previous	Save Precheck

Note Initial synchronization includes initial schema synchronization and initial full data synchronization. If you select both Initial Schema Synchronization and Initial Full Data Synchronization, DTS synchronizes the schemas and historical data of the required objects before DTS synchronizes incremental data.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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



## 7.2. Synchronize data from a selfmanaged MySQL database hosted on ECS to a PolarDB for MySQL cluster

PolarDB is a next-generation relational database service that is developed by Alibaba Cloud. PolarDB is compatible with the MySQL database engine and features high availability, ease of use, and reliability. This topic describes how to synchronize data from a self-managed MySQL database hosted on Elastic Compute Service (ECS) to a PolarDB for MySQL cluster by using Data Transmission Service (DTS).

#### Prerequisites

A PolarDB for MySQL cluster is created. For more information, see Create a 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 loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.

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

## SQL operations that can be synchronized

## Limits

- Incompatibility with triggers
   If you select a database as the object to be synchronized and the database contains a trigger that
   updates a table, data inconsistency may occur. For more information about how to solve this issue,
   see Configure a data synchronization task for a source database that contains a trigger.
- Limits on RENAME TABLE operations

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

## Before you begin

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

**Note** The database account must have the SELECT permission on the objects to be synchronized, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.

## 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, see Synchronization topologies.

## Procedure

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

**Note** On the buy page, set Source Instance to MySQL, set Destination 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 lease lease in the section of the sectin of the section of the section of the section of	1.Configure Source and	Destination	2.Select Objects to Synchronize	>	3.Advanced Settings	>	4.Precheck
Section Parameter   Section Section   N/A Synchronization   N/A Synchronization   Section Synchronization   Section Synchronization   N/A Sintance   Section Section Laboration   N/A Sintance   Section Sintance   Section Sintance   Section Sintance   Section Sintance   Section Sintance   Section Sintance   Sintance Sintance   Sintanc							
seve tenered teners          Interest Type       Service of Control telest Service Of Contr	Synchronization T	ask Name: MySQL_TO_PolarDE					
bit with the bit with the bit with the bit here is a set to MySQL and cannot be changed         • Status here is with the bit here is a set to MySQL and cannot be changed.         • Other here is a set to MySQL and cannot be changed.         • Other here is a set to MySQL and cannot be changed.         • Other here is a set to MySQL and cannot be changed.         • Other here is a set to MySQL and cannot be changed.         • Other here is a set to MySQL and cannot be changed.         • Other here is a set to MySQL and cannot be changed.         • Other here is a set to MySQL and cannot be changed.         • Other here is a set to MySQL and cannot be changed.         • Other here is a set to MySQL and cannot be changed.         • Other here is a set to MySQL and cannot be changed.         • Other here is a set to MySQL and cannot be changed.	Source Instance Details						
Interne The law of weak of backase in ECS lands are in a first and in a first an							
Image: Section of Control of the Section of Control of the Section of the Sectin of the Section of the Section of the Section	Insta	nce Type: User-Created Datab	ase in ECS Instance	٣			
Section Parameter   Synchronization Bescription   N/A Synchronization   N/A Synchronization   Section Parameter   Synchronization Bescription   N/A Synchronization   Section Parameter   Section Synchronization   N/A Synchronization   Section Section   DTS automatically generates a task name. We recommend that you selected on the bury page. You cannot be changed.   N/A Son Section   Instance Select User-Created Database in ECS Instance.   Figure Select the ID of the ECS instance on which the self-managed MySQL database.   Database This parameter is set to MySQL and cannot be changed.   Port Number Ether the service port number of the self-managed MySQL database.	* ECS In	stance ID:					
Image: Second Secon	Datab	ase Type: MySQL		Ť			
• backets Heart       • ended         • backets       • ended         •	* Por	t Number: 3306					
• 'otdute: Howe''       • 'otdute: Howe''         Detention: Howe''       Interm: Type: Marcial Interm: Type: Marcial Interm: Type: Marcial Interm: Type: Marcial Interm: Type: Interm: Type: Type: I	* Database	Account: dtstest					
Detection Indiana Detains       Indiana Type: Padd0         Indiana Type: Padd0       Indiana Type: Padd0         Indiana Type: Padd0       Indiana         Section       Parameter       Description         N/A       Synchronizati ion Task       DTS automatically generates a task name. We recommend that you specify an informative name to identify the task. You do not need to use a unique task name.         N/A       Instance Type       Select User-Created Database in ECS Instance.         Instance Region       The source region that you selected on the buy page. You cannot change the value of this parameter.         ECS Instance Instance       Select the ID of the ECS instance on which the self-managed MySQL database is deployed.         Database Type       This parameter is set to MySQL and cannot be changed.         Port Number       Enter the service port number of the self-managed MySQL databases.	* Database	Password:		Ф			
Detention Regist       Failable         Litteres Regist       Segress         * Volations Regist       DTS automatically generates a task name. We recommend that you specify an informative name to identify the task. You do not need to use a unique task name.         N/A       Instance Type       Select User-Created Database in ECS Instance.         Region       The source region that you selected on the buy page. You cannot change the value of this parameter.         ECS Instance       Select the ID of the ECS instance on which the self-managed MySQL         Database       This parameter is set to MySQL and cannot be changed.         Port Number       Enter the service port numbe							
Instance Free: Paired       Image: Support         * Paired B Interne B Image: Support       Image: Support         * Detables Accesse Image: Imag	Destination Instance Details						
Instance Type       Select User-Created Database in ECS Instance.         Instance Type       Select the ID of the ECS instance on which the self-managed MySQL         Instance Type       Select the ID of the ECS instance on which the self-managed MySQL         Instance Type       Select the ID of the ECS instance on which the self-managed MySQL         Instance Type       Select the ID of the ECS instance on which the self-managed MySQL         Instance Type       Select the ID of the ECS instance on which the self-managed MySQL         Instance Database       This parameter is set to MySQL and cannot be changed.         Instance Database       This parameter is not the self-managed MySQL database.	Insta	nce Type: PolarDB					
• Value Notations Account       • value Notations         • Value Notations       • value Notations         • Value Notations       • value Notations         • Database Notations       • value Notations         • Database Notations       • value Notations         Section       Parameter       Description         N/A       Synchronization Task Name       DTS automatically generates a task name. We recommend that you use a unique task name.         N/A       Instance Type       select User-Created Database in ECS Instance.         Instance Region       The source region that you selected on the buy page. You cannot change the value of this parameter.         ECS Instance ID       Select the ID of the ECS instance on which the self-managed MySQL         Database Type       This parameter is set to MySQL and cannot be changed.         Port Number       Inter the service port number of the self-managed MySQL database.	Instanc	e Region: Singapore					
• 'utdate Accelf:       detet         • Database Reserved:	* PolarDB In:	stance ID: pc-gs		•			
• Detables Pearwerk       •         • Detables Pearwerk       •         Central <thcentral <="" th="">       Central</thcentral>	* Database	Account: dtstest					
Section       Parameter       Description         N/A       Synchronization Task Name       DTS automatically generates a task name. We recommend that you specify an informative name to identify the task. You do not need to use a unique task name.         N/A       Instance Type       Select User-Created Database in ECS Instance.         Instance Region       The source region that you selected on the buy page. You cannot change the value of this parameter.         ECS Instance ID       Select the ID of the ECS instance on which the self-managed MySQL         Database Type       This parameter is set to MySQL and cannot be changed.         Port Number       Enter the service port number of the self-managed MySQL database.	* Database	Password:		<b>4</b> >			
Cerce       Cerce <th< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td></th<>							
SectionParameterDescriptionN/ASynchronizat ion Task NameDTS automatically generates a task name. We recommend that you specify an informative name to identify the task. You do not need to use a unique task name.Instance TypeSelect User-Created Database in ECS Instance.Instance RegionThe source region that you selected on the buy page. You cannot change the value of this parameter.ECS Instance IDSelect the ID of the ECS instance on which the self-managed MySQL database is deployed.Database TypeThis parameter is set to MySQL and cannot be changed.Port NumberEnter the service port number of the self-managed MySQL database.						Car	cel Set Whitelist and Next
SectionParameterDescriptionN/ASynchronizat ion Task NameDTS automatically generates a task name. We recommend that you specify an informative name to identify the task. You do not need to use a unique task name.Instance TypeSelect User-Created Database in ECS Instance.Instance RegionThe source region that you selected on the buy page. You cannot change the value of this parameter.ECS Instance IDSelect the ID of the ECS instance on which the self-managed MySQL database is deployed.Database TypeThis parameter is set to MySQL and cannot be changed.Port NumberEnter the service port number of the self-managed MySQL database.							
N/ASynchronization Task NameDTS automatically generates a task name. We recommend that you specify an informative name to identify the task. You do not need to use a unique task name.Instance TypeSelect User-Created Database in ECS Instance.Instance RegionThe source region that you selected on the buy page. You cannot change the value of this parameter.ECS Instance IDSelect the ID of the ECS instance on which the self-managed MySQL database is deployed.Database TypeThis parameter is set to MySQL and cannot be changed.Port NumberEnter the service port number of the self-managed MySQL database.	Section	Parameter	Description				
N/A       Drs automaticativ generates a task name. We recommend that you specify an informative name to identify the task. You do not need to use a unique task name.         Instance       Instance         Type       Select User-Created Database in ECS Instance.         Instance       The source region that you selected on the buy page. You cannot change the value of this parameter.         ECS Instance       Select the ID of the ECS instance on which the self-managed MySQL database is deployed.         Database       This parameter is set to MySQL and cannot be changed.         Port Number       Enter the service port number of the self-managed MySQL database.				llurgener	stor a tack name )		and that you
Name       Spearly an introduction of the basis of a bink you do not need to use a unique task name.         Instance       Instance         Type       Select User-Created Database in ECS Instance.         Instance       The source region that you selected on the buy page. You cannot change the value of this parameter.         ECS Instance       Select the ID of the ECS instance on which the self-managed MySQL database is deployed.         Database       This parameter is set to MySQL and cannot be changed.         Port Number       Enter the service port number of the self-managed MySQL database.	N/A	ion Task	specify an inform	native n	ares a rask name. I	task. You d	o not need to
Instance TypeSelect User-Created Database in ECS Instance.Instance RegionThe source region that you selected on the buy page. You cannot change the value of this parameter.ECS Instance IDSelect the ID of the ECS instance on which the self-managed MySQL database is deployed.Database TypeThis parameter is set to MySQL and cannot be changed.Port NumberEnter the service port number of the self-managed MySQL database.		Name	use a unique tas	k name.			
Instance TypeSelect User-Created Database in ECS Instance.Instance RegionThe source region that you selected on the buy page. You cannot change the value of this parameter.ECS Instance IDSelect the ID of the ECS instance on which the self-managed MySQL database is deployed.Database TypeThis parameter is set to MySQL and cannot be changed.Port NumberEnter the service port number of the self-managed MySQL database.							
TypeInstance RegionThe source region that you selected on the buy page. You cannot change the value of this parameter.ECS Instance IDSelect the ID of the ECS instance on which the self-managed MySQL database is deployed.Database TypeThis parameter is set to MySQL and cannot be changed.Port NumberEnter the service port number of the self-managed MySQL database.		Instance	Select User-Cre	ated D	atabase in ECS In	stance.	
Instance RegionThe source region that you selected on the buy page. You cannot change the value of this parameter.ECS Instance IDSelect the ID of the ECS instance on which the self-managed MySQL database is deployed.Database TypeThis parameter is set to MySQL and cannot be changed.Port NumberEnter the service port number of the self-managed MySQL database.		Туре					
Regionchange the value of this parameter.ECS Instance IDSelect the ID of the ECS instance on which the self-managed MySQL database is deployed.Database TypeThis parameter is set to MySQL and cannot be changed.Port NumberEnter the service port number of the self-managed MySQL database.		Instance	The source region	on that y	ou selected on the	buy page. Y	'ou cannot
ECS Instance IDSelect the ID of the ECS instance on which the self-managed MySQL database is deployed.Database TypeThis parameter is set to MySQL and cannot be changed.Port NumberEnter the service port number of the self-managed MySQL database.		Region	change the value	e of this	parameter.	<i>,</i> ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
ECS Instance IDSelect the ID of the ECS instance on which the self-managed MySQL database is deployed.Database TypeThis parameter is set to MySQL and cannot be changed.Port NumberEnter the service port number of the self-managed MySQL database.							
ID       database is deployed.         Database       This parameter is set to MySQL and cannot be changed.         Port Number       Enter the service port number of the self-managed MySQL database.		ECS Instance	Select the ID of	the ECS	nstance on which t	he self-man	aged MySQL
Database TypeThis parameter is set to MySQL and cannot be changed.Port NumberEnter the service port number of the self-managed MySQL database.		U	uarabase is dep	ioyea.			
Type       This parameter is set to MySQL and cannot be changed.         Port Number       Enter the service port number of the self-managed MySQL database.		Database	This second to the		McCOL and any		
Port Number Enter the service port number of the self-managed MySQL database.		Туре	This parameter	is set to	MYSQL and cannot	be changed	1.
Port Number Enter the service port number of the self-managed MySQL database.		Device	Estand 1				
		Port Number	Enter the service	e port nu	mper of the self-m	ianaged MyS	QL database.

Source Instance

Details

Section	Parameter	Description
		Enter the account of the self-managed MySQL database.
	Dat abase Account	<b>Note</b> The database account must have the SELECT permission on the objects to be synchronized, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.
	Dat abase Password	Enter the password of the database account.
	Instance Type	This parameter is set to <b>PolarDB Instance</b> and cannot be changed.
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.
Destination	PolarDB Instance ID	Select the ID of the destination PolarDB cluster.
Instance Details		Enter the database account of the destination PolarDB cluster.
	Database Account	<b>Note</b> The database account must have the ALL permission on the objects to be synchronized.
	Database Password	Enter the password of the database account.

7.

8. Select the processing mode of conflicting tables and the objects to be synchronized.

Synchronization Mode: <b>One-Wa</b>	y Synchronization (DML+DDL)			
Available Expand the tree before you perform therecycle_bin the	rm a glol Q 2-0001	Selected (To edit an object name of Edit.) Learn more.	r its filter, hover over the object and cl	ick
Select All  *Rename Databases and Tables: *Source table DMS_ONLINE_Do yo want to copy the temporary table to the target database during DDL: * Retry Time for Failed Connection	Do Not Change Database and Table Names u Yes No	Select All Change Database and Table Na	mes	
			Cancel Previous	Next

Setting	Description
Select the processing mode of conflicting tables	<ul> <li>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.</li> <li>Note You can use the object name mapping feature to rename the tables that are synchronized to the destination database. You can use this feature if the source and destination databases contain identical table names and the tables in the destination database cannot be deleted or renamed. For more information, see Rename an object to be synchronized.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>During initial data synchronization, DTS does not synchronize the data records that have the same primary keys as the data records that have the same scient. However, DTS synchronizes these data records during incremental data synchronization.</li> <li>If the source and destination databases have different schemas, initial data synchronized or the data synchronized or the data synchronized in the data synchronized or the data synchronized or the data synchronized in tabases have the same schema. However, DTS synchronizes these data records during incremental data synchronization.</li> </ul>
Select the objects to be synchronized	<ul> <li>Select one or more objects from the Available section and click the &gt; icon to move the objects to the Selected section. You can select tables or databases as the objects to be synchronized.</li> <li>Note <ul> <li>If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination cluster, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination, see Rename an object to be synchronized.</li> </ul> </li> </ul>

Setting	Description				
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.				
Replicate Temporary Tables When DMS Performs DDL Operations	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> <li>Note If online DDL operations generate a large amount of data, the data synchronize the data of temporary tables generated by online DDL operation task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> <li>Note If you select No, the tables in the destination database may be locked.</li> </ul>				
Retry Time for Failed Connections	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails. <b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.				

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

1.Configure Source and Destination	2.Select	Objects to Synchro	onize 🔶 .	3.Advanced Settings		4.Precheck	
Initial Synchronization: 🗹	Initial Schema Synch	ronization 🗹 Init	ial Full Data Synchronization				
				Cance	Previous	Save	Precheck

Onte Initial synchronization includes initial schema synchronization and initial full data synchronization. If you select both Initial Schema Synchronization and Initial Full Data Synchronization, DTS synchronizes the schemas and historical data of the required objects before DTS synchronizes incremental data.

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

? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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



## 7.3. Synchronize data from a selfmanaged MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to a PolarDB for MySQL cluster

PolarDB is a next-generation relational database service that is developed by Alibaba Cloud. PolarDB is compatible with the MySQL database engine and features high availability, ease of use, and reliability. This topic describes how to synchronize data from a self-managed MySQL database that is connected over Express Connect, VPN Gateway, or Smart Access Gateway to a PolarDB for MySQL cluster by using Data Transmission Service (DTS).

## Prerequisites

- The version of the self-managed MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The self-managed MySQL database is connected to a virtual private cloud (VPC) over Express Connect, VPN Gateway, or Smart Access Gateway. For more information, see Connect an on-premises database to Alibaba Cloud.

Onte DTS must be allowed to access the VPC that is connected to the self-managed MySQL database. For more information, see Configure a route between DTS and Express Connect, VPN Gateway, or Smart Access Gateway.

• A PolarDB for MySQL cluster is created. For more information, see Create a PolarDB for MySQL cluster.

(?) Note The available storage space of the PolarDB for MySQL cluster is larger than the total size of the data in the self-managed MySQL database.

### Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- •
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source 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

## 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, see Synchronization topologies.

#### Limits

• Incompatibility with triggers

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

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

## Before you begin

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

(?) Note The database account must have the SELECT permission on the objects to be synchronized, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.

### Procedure

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

**Note** On the buy page, set Source Instance to MySQL, set Target Instance to 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.

1.Configure Source and	Destination 2	Select Objects to Synchronize	>	3.Advanced Settings	>	4.	Precheck
Synchronization Ta	ask Name: MySQL_TO_PolarDB						
Source Instance Details							
Insta	nce Type: User-Created Databa	se Connected Over Express Connect, V	PN Gatev 🔻				
Instanc	e Region: Singapore						
•	Peer VPC: vpc-t4n	100	•	Proprietary network of Other Apsara Stack	Accounts		
Datab	ase Type: MySQL						
* IP	Address: 172.16.						
* Port	Number: 3306						
* Database	Account: dtstest						
* Database F	Password:		4>				
Destination Instance Details							
Insta	nce Type: PolarDB						
Instanc	e Region: Singapore						
* PolarDB Ins	tance ID: pc-g		•				
* Database	Account: dtstest						
* Database F	Password:		¢>				
						Cancel	Set Whitelist and Next
Section	Parameter	Description					

Section	Parameter	Description
N/A	Synchroniz at ion Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
Source	lnstance Type	Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway.
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.
	Peer VPC	Select the ID of the VPC that is connected to the self-managed MySQL database.
	Database Type	This parameter is set to <b>MySQL</b> and cannot be changed.
Details	IP Address	Enter the server IP address of the self-managed MySQL database.
	Port Number	Enter the service port number of the self-managed MySQL database.
	Database Account	Enter the account that you created for the self-managed MySQL database. For more information, see Before you begin.
	Database Password	Enter the password of the database account.
	lnstance Type	This parameter is set to <b>PolarDB</b> and cannot be changed.
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.
	PolarDB Instance ID	Select the ID of the destination PolarDB cluster.
Destination		Enter the database account of the destination PolarDB cluster.
Details	Database	<b>Note</b> The database account must have the ALL permission on the objects to be synchronized.
	Dat abase Password	Enter the password of the 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 whitelist of the destination PolarDB cluster. This ensures that DTS servers can connect to the destination PolarDB cluster.

8. Select the processing mode of conflicting tables and the objects to be synchronized.

1.Select Source and Destination	2.Select Object to Be Synchronized	3.Advanced Settings	4.Precheck
Synchronization Mode:One-Way         Available         Expand the tree before you perform         1 mrecycle_bin         2 mrecycle_bin         2 mrecycle_bin         2 mrecycle_bin         2 mrecycle_bin         2 mrecycle_bin         2 mrecycle_bin         3 mrecycle_bin         3 mrecycle_bin         4 mrecycle_bin         4 mrecycle_bin         5 mrecycle_bin         5 mrecycle_bin         5 mrecycle_bin         6 mrecycle_bin         6 mrecycle_bin         7 m	y Synchronization (DML+DDL)	S.Advanced Settings	filter, hover over the object and click
Select All  Rename Databases and Tables:  Source table DMS_ONLINE_Do you want to copy the temporary table to the target database during DDL:  Retry Time for Failed Connection	<ul> <li>Do Not Change Database and Table Names</li> <li>u Yes No ?</li> <li>720 Minutes ?</li> </ul>	Select All Change Database and Table Names	Cancel Previous Next
Setting	Description		

Setting	Description
Select the processing mode of conflicting tables	<ul> <li>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.</li> <li>Note You can use the object name mapping feature to rename the tables that are synchronized to the destination database. You can use this feature if the source and destination databases contain identical table names and the tables in the destination database cannot be deleted or renamed. For more information, see Rename an object to be synchronized.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>Marning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks.</li> <li>During initial data synchronization, DTS does not synchronize the data records that have the same primary keys as the data records in the destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization.</li> <li>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 synchronized rot may fails.</li> </ul>
Select the objects to be synchronized	<ul> <li>Select one or more objects from the Available section and click the ) icon to move the objects to the Selected section. You can select tables or databases as the objects to be synchronized.</li> <li>Note <ul> <li>If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination cluster, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination, see Rename an object to be synchronized.</li> </ul> </li> </ul>

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

#### 10. Select the initial synchronization types.

1.Configure Source and Destination	$\rangle$	2.Select Objects to	Synchronize		3.Advanced Settings			4.Precheck	
Tailid Cardonaladian 🗖	total cut			Construction					
Initial Synchronization: 🗹	Initial Scr	iema Synchronization	Initial Full Data	Synchronization					
						Cancel	Previous	Save	Precheck

Note Initial synchronization includes initial schema synchronization and initial full data synchronization. If you select both Initial Schema Synchronization and Initial Full Data Synchronization, DTS synchronizes the schemas and historical data of the required objects before DTS synchronizes incremental data.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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



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

is a real-time online analytical processing (RT-OLAP) service that is 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 data-driven insights into your business. This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to an cluster by using Data Transmission Service (DTS). After you synchronize data, you can use AnalyticDB for MySQL to build internal business intelligence (BI) systems, interactive query systems, and real-time report systems.

#### Prerequisites

- The tables that you want to synchronize from the ApsaraDB RDS for MySQL instance contain primary keys.
- An cluster is created. For more information, see Create an cluster.
- The destination 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 loads of the database servers. 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 impact of data synchronization on 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 utilization 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 data definition language (DDL) operations on the required objects during data synchronization. Otherwise, data may fail to be synchronized.
- Due to the limits of , if the disk space usage of the nodes in an cluster reaches 80%, the cluster is locked. We recommend that you estimate the required disk space based on the objects that you want to synchronize. You must ensure that the destination cluster has sufficient storage space.
- Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to be synchronized.

## SQL operations that can be synchronized

- DDL operations: CREATE TABLE, DROP TABLE, RENAME TABLE, TRUNCATE TABLE, ADD COLUMN, DROP COLUMN, and MODIFY 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 is stopped. You can submit a or 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 permissions
ApsaraDB RDS for MySQL	The SELECT permission on the objects to be synchronized, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission
	The read and write permissions on the objects to be synchronized

#### Data type mappings

The data types of ApsaraDB RDS for MySQL and 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 schema synchronization.

## Procedure

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

On the buy page, set Source Instance to MySQL, set Target Instance to AnalyticDB 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. In the upper part of the **Synchronization Tasks** page, select the region where the data synchronization instance resides.
- 5. Find the data synchronization instance and click **Configure Task** in the Actions column.
- 6. Configure the source and destination databases.

1.Configure Source and Destination Ir	nstances 2.Authorize AnalyticDB Account	t	3.Select Objects to Synchronize	A.Precheck
Synchronization Task Name:	DDC MuCOL TO ADD MuCOL			
Synchronization rusk numer	KUS MYSQL_TO_ADB MYSQL			
Source Instance Details				
Instance Type:	RDS Instance	•		
Instance Region:	Singapore			
* Instance ID:	rm-gs	•	RDS Instances of Other Apsara Stack Accounts	
* Database Account:	dtstest			
* Database Password:	•••••	<b>♦</b> >		
Destination Instance Details				
Instance Type:	AnalyticDB			
Instance Region:	Singapore			
*Version:	2.0 ( 3.0			
* Database:	am-gs	•		
* Database Account:	dtetaet			
Database Account				
* Database Password:	•••••	4>		

Section	Parameter	Description			
N/A	Synchroniz at io n Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.			
	Instance Type	Select RDS Instance.			
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.			
	Instance ID	Select the ID of the source RDS instance.			
		Enter the database account of the source RDS instance. For information about the permissions that are required for the account, see Permissions required for database accounts.			
	Dat abase Account	<b>Note</b> If the database engine of the source RDS instance is <b>MySQL 5.5</b> or <b>MySQL 5.6</b> , you do not need to configure the <b>database account</b> or <b>database password</b> .			
Source Instance Details					

Cancel

Section	Parameter	Description		
	Database Password	Enter the password of the database account.		
	Encryption	Select <b>Non-encrypted</b> or <b>SSL-encrypted</b> . If you want to select <b>SSL-encrypted</b> , 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 ApsaraDB RDS for MySQL instance.		
		<b>Notice</b> The <b>Encryption</b> parameter is available only for regions in the Chinese mainland and the China (Hong Kong) region.		
	Instance Type	The value of this parameter is set to <b>AnalyticDB</b> and cannot be changed.		
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.		
Destination	Version	Select 3.0.		
Instance Details	Database	Select the ID of the destination cluster.		
	Dat abase Account	Enter the database account of the cluster. For information about the permissions that are required for the account, see Permissions required for database accounts.		
	Dat abase Password	Enter the password of the database account.		

7.

8. Select the synchronization policy and the objects to be synchronized.

1.Select Source and Destinati	on 🔰 2.Authorize AnalyticDB Account >	3.Select Object to Be 4.Precheck				
Initial Synchronization: 🔽 Initial Schema Synchronization						
Note: do not clean up the incremental data log generated by the source database after the DTS task is started when the DTS full task is running. If the source database cleans up the log too early, the DTS incremental task may fail						
Proccessing Mode In Existed Target Table: Pr	e-check and Intercept $ \bigcirc $ Ignore					
Merge Multi Tables: Ye	is 🖲 No					
Synchronization Type: 🗹 In 🔽 Cr	sert 🗹 Update 🗹 Delete 🗹 Alter Ta reate Table 🗹 Drop Table	ble 🗹 Truncate Table				
Available		Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.				
Expand the tree before you pe	erform a glol 🔰 🔍					
🕀 🗁recycle_bin						
🖅 🦢 asd		otstestdata				
🕀 📂 dtstest0512_jzhz_000	1_ext_0001					
🗄 🦢 dtstest123	<					
E Sys						
Select All						
Successi		Select All				
*Rename Databases and Tables:	Do Not Change Database and Table Nam	es O Change Database and Table Names				
*Source table DMS_ ONLINE_ Do want to copy the temporary table the target database during DDL:	you 💿 Yes 🔿 No 🕜 to	(F				
* Retry Time for Failed Connectio	n 720 Minutes 🧭					
Cancel Previous Next Precheck						
Parameter	Description					
Initial Synchronization	You must select both <b>Initial S</b> <b>Synchronization</b> in most case schemas and data of the requ destination cluster. The schem incremental synchronization.	chema Synchronization and Initial Full Data es. After the precheck, DTS synchronizes the ired objects from the source instance to the has and data are the basis for subsequent				

Parameter	Description			
Processing Mode In Existed Target Table	<ul> <li>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.</li> <li>Note You can use the object name mapping feature to change the names of the tables that are synchronized to the destination database. You can use this feature if the source and destination database contain identical table names and the tables in the destination database cannot be deleted or renamed. For more information, see Rename an object to be synchronized.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>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.</li> <li>If the source and destination databases have different schemas, initial data synchronized or the data synchronization task fails.</li> </ul>			
Merge Multi Tables	<ul> <li>If you select Yes, DTS adds thedts_data_source column to each table to record data sources. In this case, DDL operations cannot be synchronized.</li> <li>No is selected by default. In this case, DDL operations can be synchronized.</li> <li>Note You can merge the data source columns based on tasks rather than tables. To merge only the data source columns of specific tables, you can create two data synchronization tasks.</li> </ul>			
Synchronization Type	Select the types of operations that you want to synchronize based on your business requirements. All operation types are selected by default. For more information, see SQL operations that can be synchronized.			

Parameter	Description			
	Select one or more objects from the <b>Available</b> section and click the > icon to move the objects to the <b>Selected</b> section. You can select tables or databases as the objects to be synchronized.			
Select the objects to be synchronized	<ul> <li>Note</li> <li>If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.</li> <li>If you select a table as the object to be synchronized, only the ADD COLUMN operations that are performed on the table are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination cluster, the name of the object remains unchanged. You can use the object name mapping feature to change the names of the objects that are synchronized to the destination cluster. For more information, see Rename an object to be synchronized.</li> </ul>			
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.			
Replicate Temporary Tables When DMS Performs DDL Operations	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DD operations.</li> <li>Note If online DDL operations generate a large amount of data, the data synchronization task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> <li>Note If you select No, the tables in the destination database may be locked.</li> </ul>			

Parameter	Description
Retry Time for Failed Connections	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

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

10. Specify a type for the tables that you want to synchronize to the destination database.

1.Configure Source and Destination Instances 🔪			halyticDB Account 3.Select Objects to Synchronize			
AnalyticDB Table Group	AnalyticDB Table Name	Type(All) 👻	Primary Key Column	Distribution Column	Definition Status(All) 👻	
dtstestdata	customer	Partitioned 1 🔻	id	id 🔻	Defined	
dtstestdata	order	Partitioned 1 🔻	orderid	orderid <b>v</b>	Defined	
Set All to Partitioned Table	Set All to Dimension Table Enter a table na	ame. Search		Total: 2 item(s), Per Page: 2	) v item(s) « < 1 > »	
				Cancel	Previous Save Precheck	

Once After you select Initial Schema Synchronization, you must specify the type, primary key column, and partition key column for the tables that you want to synchronize to. 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, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.
- 12. Close the **Precheck** dialog box after the following message is displayed: **Precheck Passed**. Then, the data synchronization task starts.
- 13. Wait until initial synchronization is complete and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1	> >>

# Troubleshoot the synchronization failure that occurs due to field type changes

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 is stopped. You can submit a or perform the following steps to troubleshoot the issue.

- Create a table in the destination cluster based on the schema of source table that fails to be synchronized. For example, if a table named customer (Table A) fails to be synchronized, you can create a table named customer\_new (Table B) in the destination cluster. Make sure that Table B has the same schema as Table A.
- 2. Run the INSERT INTO SELECT command to copy the data of Table A and insert the data into Table B. This ensures that the data of the two tables is consistent.
- 3. Rename or delete Table A. Then, change the name of Table B to customer.
- 4. Restart the data synchronization task in the DTS console.

# 7.5. 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 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 that you want to synchronize from the ApsaraDB RDS for MySQL instance contain primary keys.
- The destination instance is created. For more information, see Create an AnalyticDB for PostgreSQL instance.

## Precautions

DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.

## Limits

- You can select only tables as the objects to be synchronized.
- DTS does not synchronize the following types of data: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID\_SNAPSHOT.
- Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to 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.

## SQL operations that can be synchronized

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

MySQL	
Database	Schema
Table	Table

## Procedure

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

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.

1.Configure Source and Destinat	ion Instances in	2.Select Objects to Syncl	nronize	>	3.Precheo	:k
Synchronization Task Name:	MySQL_TO_ADB for PostgreSQL					
Source Instance Details						
Instance Type:	RDS Instance	Ŧ				
Instance Region:	China (Hangzhou)					
* Instance ID:	rm-bj	-	RDS Instances of Oth	er Apsara Stack Accounts		
* Database Account:	dtstest					
* Database Password:	•••••	4>				
* Encryption:	Non-encrypted OSSL-encrypted					
Destination Instance Details						
Instance Type:	ApplyticDR for PosteroSOL					
Instance Type:	China (Hanesheu)					
Instance Region:	China (Hangzhou)					
* Instance ID:	gp-bp	-				
* Database Name:	dtstestdata					
* Database Account:	dtstest					
* Database Password:	••••••	₫>				
					Cancel	Set Whitelist and Next

Section	Parameter	Description
None	Synchronizat i on Task Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.
	Instance Type	Select RDS Instance.
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.
	Instance ID	The ID of the source ApsaraDB RDS for MySQL instance.

Section	Parameter	Description
Source Instance Details	Dat abase Account	The database account of the ApsaraDB RDS for MySQL instance.
		<b>Note</b> If the database engine of the source ApsaraDB RDS for MySQL instance is <b>MySQL 5.5</b> or <b>MySQL 5.6</b> , you do not need to configure the <b>database account</b> and <b>database password</b> .
	Dat abase Password	The password of the database account.
	Encryption	Select Non-encrypted or SSL-encrypted. If you select SSL- encrypted, you must enable SSL encryption for the ApsaraDB RDS for MySQL instance before you configure the data synchronization task. For more information, see Configure SSL encryption for an ApsaraDB RDS for MySQL instance.
		regions in the Chinese mainland and the China (Hong Kong) region.
Destination Instance Details	Instance Type	The value of this parameter is set to <b>AnalyticDB for</b> <b>PostgreSQL</b> and cannot be changed.
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.
	The ID of the destination instance.	The ID of the destination instance.
	Database Name	The name of the destination database.
	Dat abase Account	The <b>initial account</b> of the instance. For more information, see <b>Create a database account</b> .
		<b>Note</b> You can also enter an account that has the RDS_SUPERUSER permission. For more information, see <b>Manage users and permissions</b> .
	Database Password	The password of the database account.

7.

8. Select the synchronization policy and the objects to be synchronized.
| .Select Source and Destination Instances for 🔰   | 2.Select Obje  | t to Be Synchronized                                     |                                       | 3.Precheck                                |      |
|--|--|--|---------------------------------------|---|------|
| Initial Synchronization: Initial Schema Synchron<br>Note: do not clean up the incremental data log genera<br>cleans up the log too early, the DTS incremental task | ization Initial Ful<br>ited by the source databa<br>may fail | l Data Synchronization<br>se after the DTS task is start | ed when the DTS full t                | ask is running. If the source database    |      |
| Proccessing Mode In<br>Existed Target Table:       Pre-check and Intercept   | $\bigcirc$ Clear Target Table $\bigcirc$                     | Ignore   |                                       |   |      |
| Synchronization Type: Insert Update  | ✓ Delete   | Alter Table  |                                       |   |      |
| Available  |  | Selected (To<br>Edit.) <u>Learn</u>                      | edit an object name o<br><u>nore.</u> | r its filter, hover over the object and o | lick |
| Expand the tree before you perform a glol $\hfill \ensuremath{\mathbf{Q}}$   |  |  |                                       | Q   |      |
| recycle_bin  |  | 💼 dtstestd   | lata                                  |   |      |
| + 🧁 chw02  |  |  |                                       |   |      |
|  |  | >  |                                       |   |      |
|  |  | -  |                                       |   |      |
| 🕀 📂 dtstestdata1   |  | <  |                                       |   |      |
| 🕂 🦢 sys  |  |  |                                       |   |      |
|  |  |  |                                       |   |      |
|  |  |  |                                       |   |      |
|  |  |  |                                       |   |      |
|  |  |  |                                       |   |      |
| Select All   |  |  |                                       |   |      |
|  |  | Select All   |                                       |   |      |
| *Rename Databases and Tables:   Do Not (   | Change Database and Tab                                      | le Names 🛛 Change D                                      | atabase and Table Na                  | mes                                       |      |
| *Source table DMS_ ONLINE_ Do you  | () No (?)  |  |                                       |   |      |
| * Retry Time for Failed Connection 720   | Minutes 🕜  |  |                                       |   |      |
| * Add quotation marks to the target<br>object  | O No   |  |                                       |   |      |
|  |  |  |                                       |   |      |

Setting	Parameter	Description
	Initial Synchronization	You must select both <b>Initial Schema Synchronization</b> and <b>Initial Full Data Synchronization</b> 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.

Setting	Parameter	Description	
Select the synchronizatio n policy	Processing Mode of Conflicting Tables	<ul> <li>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.     <li>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.     </li> </li></ul>	
	Synchronization Type	<ul> <li>Select the types of operations that you want to synchronize based on your business requirements.</li> <li>Insert</li> <li>Update</li> <li>Delete</li> <li>AlterTable</li> </ul>	
Select the objects to be synchronized	N/A	<ul> <li>Select one or more tables from the Available section and click the icon to move the tables to the</li> <li>Selected section.</li> <li>? Note <ul> <li>You can select only tables as the objects to be synchronized.</li> <li>You can use the object name mapping feature to change the names of the columns that are synchronized to the destination database. For more information, see Rename an object to be synchronized.</li> </ul> </li> </ul>	
Rename Databases and Tables	N/A	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name	

Setting	Parameter	Description	
	N/A	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> </ul>	
Replicate Temporary Tables When DMS Performs		<b>Note</b> If online DDL operations generate a large amount of data, the data synchronization task may be delayed.	
DDL Operations		• <b>No</b> : DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.	
		<b>Note</b> If you select No, the tables in the destination database may be locked.	
Retry Time for Failed Connections	N/A	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.	
		<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.	

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

1.Configure S	Source and Destination In	stances in	2.Select Objects to Synchronize	3.Precheck
Schema	Table	Primary Key Column	Distribution Column	Definition Status(All) 👻
dtstestdata	customer	id	id 💌	Defined
dtstestdata	order	orderid	orderid •	Defined
dts.migration.mess	age.greenplu Search		Т	otal: 2 item(s), Per Page: $20$ $\bullet$ item(s) $\ll$ $<$ 1 $>$ $\gg$
				Cancel Previous Save Precheck

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 Define constraints and Define table distribution.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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.

	Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
	0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
	Pause Task Delete Task			Total: 1 item(s), F	Per Page: 20 item(s)	« < 1	> >>

# 7.6. Synchronize data from a selfmanaged MySQL database hosted on ECS to an AnalyticDB for PostgreSQL instance

This topic describes how to synchronize data from a self-managed MySQL database hosted on Elastic Compute Service (ECS) to an instance by using Data Transmission Service (DTS).

# Prerequisites

- A MySQL database of version 5.1, 5.5, 5.6, 5.7, or 8.0 is created. The database is hosted on an ECS instance.
- The binary logging feature is enabled for the source database. A database account is created for the data synchronization task. For more information, see Create an account for a user-created MySQL

dat abase and configure binary logging.

(?) Note The database account must have the SELECT permission on the objects to be synchronized, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.

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

## Precautions

DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.

# Limits

- You can select only tables as the objects to be synchronized.
- DTS does not synchronize the following types of data: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID\_SNAPSHOT.
- Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to 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.

# SQL operations that can be synchronized

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

MySQL	
Database	Schema
Table	Table

# Procedure

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

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.

ion Instances in	2.Select Objects to Synchronize	>	3.Precheck
MySQL_TO_ADB for PostgreSQL			
User-Created Database in ECS Instance	Ŧ		
China (Hangzhou)			
i-bp	-		
1ySQL			
3306			
dtetaet			
utstest			
******	<i>4</i> >		
nalyticDB for PostgreSQL			
China (Hangzhou)			
gp-bp)	•		
dtstestdata			
distant			
utstest			
******	<b>4</b> >		
			Cancel Set Whitelist and Next
	on Instances in         MySQL_TO_ADB for PostgreSQL         User-Created Database in ECS Instance         hina (Hangzhou)         +bp         ySQL         3306         itstest         nalyticDB for PostgreSQL         hina (Hangzhou)         statest         itstest         statest         itstest         itstest         itstest         itstest	Instances in 2.Select Objects to Synchronize   MySQL_TO_ADB for PostgreSQL   User-Created Database in ECS Instance     User-Created Database in ECS Instance     vina (Hangzhou)     +bp   vSQL   3306   itstest   analyticDB for PostgreSQL   itstest   inia (Hangzhou) p-bp    itstest   itstest   itstest	on Instances in 2.Select Objects to Synchronize

Section	Parameter	Description	
N/A	Synchroniz ation Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.	
	lnstance Type	Select User-Created Database in ECS Instance.	
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.	
	Instance ID	Select the ID of the ECS instance that hosts the self-managed MySQL database.	
	Database Type	The value of this parameter is set to <b>MySQL</b> and cannot be changed.	
Source Instance Details	Port Number	Enter the service port number of the source database. The default port number is <b>3306</b> .	
		Enter the account of the self-managed MySQL database.	
	Dat abase Account	<b>Note</b> The database account must have the SELECT permission on the objects to be synchronized, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.	
	Dat abase Password	Enter the password of the source database account.	
	lnstance Type	The value of this parameter is set to <b>AnalyticDB for PostgreSQL</b> and cannot be changed.	
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.	
	Instance ID	Select the ID of the destination instance.	
Destination	Database Name	Enter the name of the destination database.	
Instance Details	Database	Enter the <b>initial account</b> of the instance. For more information, see Create a database account.	
	Dat abase Account	Note You can also enter an account that has the RDS_SUPERUSER permission. For more information, see Manage users and permissions.	
	Database Password	Enter the password of the destination database account.	

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

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

8. Select the synchronization policy and the objects to be synchronized.

1.Select Source and	Destination Instances for 2.Select	Object to Be Sync	hronized 3.Precheck
Initial Synchroniza	ation: 🔽 Initial Schema Synchronization 🔽 Ini	itial Full Data Synchro	nization
Note: do not clean u cleans up the log too	p the incremental data log generated by the source o early, the DTS incremental task may fail	database after the DT	'S task is started when the DTS full task is running. If the source database
Proccessing Mode In Existed Target Table:	$lacksquare$ Pre-check and Intercept $\bigcirc$ Clear Target Ta	ible 🔿 Ignore	
Synchronization 1	Type: ✔ Insert ✔ Update ✔ Delete	✓ Alter Table	
Available			Selected (To edit an object name or its filter, hover over the object and click Edit.) $\underline{\text{Learn more.}}$
Expand the tree bef	fore you perform a glol 🛛 🔍 🔍		
	_		📔 dtstestdata
entrop	ista 0001 out 0001	>	
	J2n2_0001_ext_0001	<	
🗄 🦢 dtstestdata1 관 🎥 sys			
Select All			Select All
*Rename Databases an	nd Tables:	nd Table Names	Change Database and Table Names
*Source table DMS_ Of want to copy the tempo the target database dur	VLINE_Do you		
* Retry Time for Failed	Connection 720 Minutes	0	
* Add quotation marks object	to the target <ul> <li>Yes</li> <li>No</li> </ul>		E
			Cancel Previous Next Precheck
Setting	Parameter	Description	
	Initial Synchronization	You must s and <b>Initial</b> After the pr data of the	elect both <b>Initial Schema Synchronization</b> <b>Full Data Synchronization</b> in most cases. recheck, DTS synchronizes the schemas and required objects from the source instance to

the destination instance. The schemas and data are the basis for subsequent incremental synchronization.

Setting	Parameter	Description
Select the synchronizatio n policy	Processing Mode of Conflicting Tables	<ul> <li>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.     <li>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.     </li> </li></ul>
	Synchronization Type	<ul> <li>Select the types of operations that you want to synchronize based on your business requirements.</li> <li>Insert</li> <li>Update</li> <li>Delete</li> <li>AlterTable</li> </ul>
Select the objects to be synchronized	N/A	<ul> <li>Select one or more tables from the Available section and click the &gt; icon to move the tables to the</li> <li>Selected section.</li> <li>? Note <ul> <li>You can select only tables as the objects to be synchronized.</li> <li>You can use the object name mapping feature to change the names of the columns that are synchronized to the destination database. For more information, see Rename an object to be synchronized.</li> </ul> </li> </ul>
Rename Databases and Tables	N/A	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.

Setting	Parameter	Description		
Replicate Temporary Tables When DMS Performs DDL Operations	N/A	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> <li>Yote If online DDL operations generate a large amount of data, the data synchronization task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> <li>Note If you select No, the tables in the destination database may be locked.</li> </ul>		
Retry Time for Failed Connections	N/A	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails. <b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.		

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

1.Configure S	Source and Destination Ir	nstances in	2.Select Objects to Synchronize	3.Precheck
Schema	Table	Primary Key Column	Distribution Column	Definition Status(All) 👻
dtstestdata	customer	id	id 💌	Defined
dtstestdata	order	orderid	orderid <b>v</b>	Defined
dts.migration.mess	age.greenplu Search		To	otal: 2 item(s), Per Page: 20 V item(s) « < 1 > »
				Cancel Previous Save Precheck

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 Define constraints and Define table distribution.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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.

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscription Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1 > »

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

**Note** The database account must have the SELECT permission on the objects to be synchronized, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.

 The on-premises network to which the self-managed MySQL database belongs is connected to Alibaba Cloud VPC over Express Connect, VPN Gateway, or Smart Access Gateway. DTS is allowed to access the network to which Express Connect, VPN Gateway, or Smart Access Gateway belongs. For more information, see Configure a route between DTS and Express Connect, VPN Gateway, or Smart Access Gateway.

Onte For more information about how to connect an on-premises network to a VPC, see Connect an on-premises database to Alibaba Cloud.

• The destination instance is created. For more information, see Create an AnalyticDB for PostgreSQL instance.

# Precautions

DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.

# Limits

- You can select only tables as the objects to be synchronized.
- DTS does not synchronize the following types of data: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID\_SNAPSHOT.
- Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to 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.

# SQL operations that can be synchronized

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

Onte 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

MySQL	
Database	Schema
Table	Table

# Procedure

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

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.

#### Dat a Transmission Service

#### Data Synchronization Synchronize d ata from a MySQL database to a dif ferent type of database

1.Configure Source and Destination I	Instances in Synchronization 2.Select Obj	ects to Synchronize	$\rightarrow$	3.Precheck	
Synchronization Task Name:	MySQL_TO_AnalyticDB for PostareSQL				
-,	histor_ro_Analyacoo for rosignesqu				
Source Instance Details					
Technology Trans					
Instance Type:	User-Created Database Connected Over Express Connect, VPI				
Instance Region:	China (Hangzhou)				
* Peer VPC:	vpc-bp:	Proprietary network of Other Aps	ara Stack Accounts		
Database Type:	MySQL				
* IP Address:	12.499.0				
* Port Number:	3306				
* Database Account:	dtstest				
* Database Password:	«۵				
Destination Instance Details					
Destination Instance Details					
Instance Type:	AnalyticDB for PostgreSQL				
Instance Region:	China (Hangzhou)				
* Instance ID:	gp-1u -				
* Database Name:	dtstestdata				
* Database Account:	dtstest				
* Database Password:	······				
				Cancel	Set Whitelist and Next

Section	Parameter	Description
N/A	Synchroniza tion Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
	lnstance Type	Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway.
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.
	Peer VPC	Select the ID of the VPC that is connected to the self-managed 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 self-managed MySQL database.
	Port Number	Enter the service port number of the source database. The default port number is <b>3306</b> .

Source Instance Details

Section	Parameter	Description
		Enter the account of the self-managed MySQL database.
	Dat abase Account	<b>Note</b> The database account must have the SELECT permission on the objects to be synchronized, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.
	Dat abase Password	Enter the password of the source database account.
	Instance Type	The value of this parameter is set to <b>AnalyticDB for PostgreSQL</b> and cannot be changed.
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.
	Instance ID	Select the ID of the destination instance.
Destination	Database Name	Enter the name of the destination database.
Instance Details		Enter the <b>initial account</b> of the instance. For more information, see <b>Create a database account</b> .
	Dat abase Account	<b>Note</b> You can also enter an account that has the RDS_SUPERUSER permission. For more information, see Manage users and permissions.
	Dat abase 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 whitelist of the instance. This ensures that DTS servers can connect to the destination instance.

#### 8. Select the synchronization policy and the objects to be synchronized.

Note: do not clean up the incremental cleans up the log too early, the DTS inc	data log generated by the source cremental task may fail	e database after the I	DTS task is started when the DTS full task is running. If the source d	atabase
Proccessing Mode In Existed Target Table:   Pre-check	and Intercent O Clear Target 1	Table O Tonore		
Synchronization Type: VI Insert	✓ Update ✓ Delete	Alter Table		
Available			Selected (To edit an object name or its filter, hover over the obje Edit.) Learn more.	ct and cli
Expand the tree before you perform a	glol Q		0	
recycle_bin			📑 dtstestdata	
🗄 🧰 asd			-	
🕀 🤭 dts		>		
Here and the statest marked by the statest m	0001	>		
🕀 🥯 dtstest 123				
🕂 🦢 dtstest123 🕂 🦢 dtstestdata1		<		
<ul> <li>dtstest123</li> <li>dtstestdata1</li> <li>sys</li> </ul>		<		
<ul> <li>dstest123</li> <li>dstestdata1</li> <li>sys</li> </ul>		<		
Itstest123 Itstestdata1 Isys		<		
<ul> <li>dstest123</li> <li>dstestdata1</li> <li>sys</li> </ul>		<		
<pre></pre>		<		
<ul> <li>dstest123</li> <li>dstestdata1</li> <li>sys</li> </ul>		<	Select All	
dtstest123     dtstestdata1     sys  Select All  Rename Databases and Tables:	Do Not Change Database	and Table Names	Select All O Change Database and Table Names	
e dtstest123 dtstest123 dtstestdata1 sys sys select All *Rename Databases and Tables: *Source table DMS_ ONLINE_ Do you want to copy the temporary table to the target database during DDL:	<ul> <li>Do Not Change Database</li> <li>Yes O No ?</li> </ul>	and Table Names	Select All Change Database and Table Names	
e dtstest123 dtstest123 dtstestdata1 sys select All *Rename Databases and Tables: *Source table DMS_ONLINE_ Do you want to copy the temporary table to the target database during DDL: * Retry Time for Failed Connection	Do Not Change Database     Yes No ?	and Table Names	Select All <ul> <li>Change Database and Table Names</li> </ul>	

Setting	Parameter	Description
	Initial Synchronization	You must select both <b>Initial Schema Synchronization</b> and <b>Initial Full Data Synchronization</b> 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.

Setting	Parameter	Description
Select the synchronizatio n policy	Processing Mode of Conflicting Tables	<ul> <li>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.     <li>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 tables to one table, you can select this mode.     </li> </li></ul>
	Synchronization Type	<ul> <li>Select the types of operations that you want to synchronize based on your business requirements.</li> <li>Insert</li> <li>Update</li> <li>Delete</li> <li>AlterTable</li> </ul>
Select the objects to be synchronized	N/A	<ul> <li>Select one or more tables from the Available section and click the icon to move the tables to the</li> <li>Selected section.</li> <li>Note <ul> <li>You can select only tables as the objects to be synchronized.</li> <li>You can use the object name mapping feature to change the names of the columns that are synchronized to the destination database. For more information, see Rename an object to be synchronized.</li> </ul> </li> </ul>
Rename Databases and Tables	N/A	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.

Parameter	Description
N/A	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> </ul>
	<b>Note</b> If online DDL operations generate a large amount of data, the data synchronization task may be delayed.
	• <b>No</b> : DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.
	<b>Note</b> If you select No, the tables in the destination database may be locked.
	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
N/A	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.
	Parameter N/A

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

1.Configure S	Source and Destination In	stances in	2.Select Objects to Synchronize	3.Precheck
Schema	Table	Primary Key Column	Distribution Column	Definition Status(All) 👻
dtstestdata	customer	id	id 💌	Defined
dtstestdata	order	orderid	orderid <b>v</b>	Defined
dts.migration.mess	age.greenplu Search		١	Total: 2 item(s), Per Page: $20$ $\bullet$ item(s) $\ll$ $\langle$ 1 $\rightarrow$ $\gg$
				Cancel Previous Save Precheck

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 Define constraints and Define table distribution.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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.

	Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
	0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
	Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1	> >>

# 7.8. Synchronize data from an ApsaraDB RDS for MySQL instance to a DataHub instance

DataHub is a real-time data distribution platform that is designed to process streaming data. You can publish and subscribe to streaming data in DataHub and distribute the data to other platforms. DataHub allows you to analyze streaming data and build applications based on streaming data. This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to a DataHub instance by using Data Transmission Service (DTS). After you synchronize data, you can use big data services such as Realtime Compute for Apache Flink to analyze the data in real time.

# Prerequisites

- The DataHub instance resides in the China (Hangzhou), China (Shanghai), China (Beijing), or China (Shenzhen) region.
- A DataHub project is created to receive the synchronized data. For more information, see Create a

#### project.

• The tables to be synchronized from the ApsaraDB RDS for MySQL instance have PRIMARY KEY or UNIQUE constraints.

# Limits

- Initial full data synchronization is not supported. DTS does not synchronize historical data of the required objects from the source RDS instance to the destination DataHub instance.
- Only tables can be selected as the objects to be synchronized.
- After a data synchronization task is started, DTS does not synchronize columns that are created in the source RDS instance to the destination DataHub instance.
- We recommend that you do not perform data definition language (DDL) operations on the required objects during data synchronization. Otherwise, data synchronization may fail.

# SQL operations that can be synchronized

INSERT, UPDATE, and DELETE

# Procedure

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

**?** Note On the buy page, set Source Instance to MySQL, set Destination Instance to DataHub, 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.

1.Specify Source and	Destination Dat	abases >	2.Select Object to Be Synchronized	3.Precheck
Synchroniz	ation Task Name:	hangzhou-h	iangzhou-medium	]
Source Instance Details				
	Instance Type:	RDS Instan		
	Instance Region:	China (Hangz	thou)	
	* Instance ID:	rm	-	RDS Instances of Other Apsara Stack Accounts
* Di	atabase Account:	dtstest		
* Dat	abase Password:	•••••	₫>	
	* Encryption:	Non-encry	vpted OSSL-encrypted	
Destination Instance Deta	ails			
	Instance Type: Instance Region:	DataHub China (Hangz	(hou)	
	* Project:	dtstestdata	•	
				Cancel Set Whitelist and Next
Section	Paramete	er	Description	
N/A	Synchron n Task No	izatio ame	DTS automatically generates a task specify an informative name to iden to use a unique task name.	name. We recommend that you tify the task. You do not need
			Select an instance type based on th database. In this example, select RD	e deployment of the source OS Instance.
	Instance	Туре	<b>Note</b> If your source databated database, you must deploy the new source database. For more inform <b>overview</b> .	ase is a self-managed MySQL etwork environment for the nation, see <mark>Preparation</mark>
	Instance Region		The source region that you selected change the value of this parameter.	on the buy page. You cannot
	Instance	ID	Select the ID of the source RDS insta	ance.
Source Instance Details	Dat abase Account	2	Enter the database account of the s <b>Note</b> If the database engir is MySQL 5.5 or MySQL 5.6, you database account or database	ource RDS instance. ne of the source RDS instance do not need to configure the password.

Section	Parameter	Description					
	Dat abase Password	Enter the password of the database account.					
	Encryption	Select <b>Non-encrypted</b> or <b>SSL-encrypted</b> . If you want to select <b>SSL-encrypted</b> , 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 ApsaraDB RDS for MySQL instance.					
		<b>Notice</b> The Encryption parameter is available only for regions in the Chinese mainland and the China (Hong Kong) region.					
Destination	Instance Type	The value of this parameter is set to <b>DataHub</b> and cannot be changed.					
Instance Details	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.					
	Project	Select the name of the DataHub <b>project</b> .					

7.

8. Select the synchronization policy and the objects to be synchronized.

1.Specify Source and Destina	tion Databases 2.Select Object to	Be Synchronized 3.Precheck
Initial Synchronization: 🔽 I	Initial Schema Synchronization	
Available		Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.
Expand the tree before you per	<pre>rform a glol Q  1_ext_0001 _0001  O Not Change Database and Table Nar n 720 Minutes ? al O Yes No</pre>	Edit.) Learn more.
column rules		
		Cancel Previous Precheck
Setting	Description	
Initial Synchronization	Select Initial Schema Synchro <b>Note</b> After you select synchronizes the schemas of destination DataHub instance	onization. Initial Schema Synchronization, DTS the required objects (such as tables) to the e.
Select the objects to be synchronized	Select one or more objects from move the objects to the Select <b>Note</b>	m the Available section and click the > icon to ted section. Tables as the objects to be synchronized. Tables as the object to the destination of the object remains unchanged. You can use oping feature to rename the objects that are destination instance. For more information, see the synchronized.

Setting	Description
Whether to enable the new naming rules for additional	After DTS synchronizes data to DataHub, DTS adds additional columns to the destination topic. If the names of additional columns are the same as the names of existing columns in the destination topic, data synchronization fails. Select <b>Yes</b> or <b>No</b> to specify <b>whether you want to enable the new naming rules for additional columns</b> .
columns	<ul> <li>Warning Before you specify this parameter, check whether additional columns and existing columns in the destination topic have name conflicts.</li> <li>For more information, see Modify the naming rules for additional columns.</li> </ul>
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.
	If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.
	• Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.
Replicate Temporary Tables When DMS	<b>Note</b> If online DDL operations generate a large amount of data, the data synchronization task may be delayed.
Performs DDL Operations	• <b>No</b> : DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.
	<b>Note</b> If you select No, the tables in the destination database may be locked.
Petry Time for	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

9. (Optional)In the **Selected** section, move the pointer over the destination topic and click **Edit**. In the dialog box that appears, set the shard key. The shard key is used for partitioning.

Available	Selected (To edit an object name or its filter, hover over the object and Edit.) Learn more.
Expand the tree before you perform a glol	Q
🕂 🚞 asd	
+ mathematical testexcel	distestdata (100jects)
+ 🖕 test123	
+ = test12345	
+ hot_dtstest0512_jzhz_0001	
+ _ test_polar2	
±) modestestdata	
etitest	
asstest	
t → → orderinfo →	
ielect All	
	Select All
tename Databases and Tables: <ul> <li>Do Not Change Database and Table Names</li> </ul>	<ul> <li>Change Database and Table Names</li> </ul>
Retry Time for Failed Connection 720 Minutes 🥥	

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the 🕡 icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscription Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1 > »

# Schema of a DataHub topic

When DTS synchronizes incremental data to a DataHub topic, DTS adds additional columns to store metadata. The following figure shows the schema of a DataHub topic.

(?) Note In this example, id, name, address are data fields. DTS adds the dts\_ prefix to data fields because the previous naming rules for additional columns are used.

dts_id	dts_name	dts_address	dts_record_id	dts_operation_flag	dts_instance_id	dts_db_name	dts_table_name	dts_utc_timestamp	dts_before_flag	dts_after_flag
10006	- <b>1</b> 11		1574832130000000000	U		dtstestdata	customer	1574832130	Y	N
10006	- 1810 - I		1574832130000000000	U		dtstestdata	customer	1574832130	N	Y
10009	1822	10010	1574832919000000000	D		dtstestdata	customer	1574832919	Y	N
10112	100		1574832919000000000	1		dtstestdata	customer	1574832919	N	Y

Previous New additional column additional Data type Description name column name The unique ID of the incremental log entry. ? Note • By default, the ID autoincrements for each new log entry. In disaster recovery scenarios, rollback may occur, and the ID may not auto-increment. Therefore, dts record new dts sync dts rec some IDs may be duplicated. String ord id \_id • If an UPDATE operation is performed, DTS generates two incremental log entries to record the pre-update and post-update values. The values of the dts record id field in the two incremental log entries are the same.

The following table describes the additional columns in the DataHub topic.

Previous additional column name	New additional column name	Data type	Description
dts_operat ion_flag	new_dts_sync_dts_ope ration_flag	String	<ul> <li>The operation type. Valid values:</li> <li>I: an INSERT operation</li> <li>D: a DELETE operation</li> <li>U: an UPDATE operation</li> </ul>
dts_instan ce_id	<pre>new_dts_sync_dts_ins tance_id</pre>	String	The server ID of the database. The value is set to null . To ensure database security, the actual value is not displayed.
dts_db_nam	new_dts_sync_dts_db_ name	String	The name of the database.
dts_table_ name	<pre>new_dts_sync_dts_tab le_name</pre>	String	The name of the table.
dts_utc_ti mestamp	<pre>new_dts_sync_dts_utc _timestamp</pre>	String	The operation timestamp, in UTC. It is also the timestamp of the binary log file.
dts_before _flag	<pre>new_dts_sync_dts_bef ore_flag</pre>	String	Indicates whether the column values are pre-update values. Valid values: Y and N.
dts_after_ flag	<pre>new_dts_sync_dts_aft er_flag</pre>	String	Indicates whether the column values are post-update values. Valid values: Y and N.

# Additional information about the dts\_before\_flag and dts\_after\_flag fields

The values of the dts\_before\_flag and dts\_after\_flag fields in an incremental log entry vary with different operation types:

INSERT

For an INSERT operation, the column values are the newly inserted record values (post-update values). The value of the dts\_before\_flag field is N, and the value of the dts\_after\_flag field is Y.

dts_id	dts_name	dts_address	dts_record_id	dts_operation_flag	dts_instance_id	dts_db_name	dts_table_name	dts_utc_timestamp	dts_before_flag	dts_after_flag
10112	194		1574832919000000000	I.		dtstestdata	customer	1574832919	N	Y

• UPDATE

DTS generates two incremental log entries for an UPDATE operation. The two incremental log entries have the same values for the <code>dts\_record\_id</code>, <code>dts\_operation\_flag</code>, and <code>dts\_utc\_timestamp</code> fields.

The first log entry records the pre-update values. Therefore, the value of the dts\_before\_flag field is Y, and the value of the dts\_after\_flag field is N. The second log entry records the post-update values. Therefore, the value of the dts\_before\_flag field is N, and the value of the dts\_after\_flag field is Y.

dts_id	dts_name	dts_address	dts_record_id	dts_operation_flag	dts_instance_id	dts_db_name	dts_table_name	dts_utc_timestamp	dts_before_flag	dts_after_flag
10006	100		1574832130000000000	U		dtstestdata	customer	1574832130	Y	N
10006	100		1574832130000000000	U		dtstestdata	customer	1574832130	N	Y

#### • DELETE

For a DELETE operation, the column values are the deleted record values (pre-update values). The value of the dts\_before\_flag field is Y, and the value of the dts\_after\_flag field is N.

dts_id	dts_name	dts_address	dts_record_id	dts_operation_flag	dts_instance_id	dts_db_name	dts_table_name	dts_utc_timestamp	dts_before_flag	dts_after_flag
10009	100	10010	1574832919000000000	D		dtstestdata	customer	1574832919	Y	N

# What to do next

After you configure the data synchronization task, you can use Realtime Compute for Apache Flink to analyze the data that is synchronized to the DataHub instance. For more information, see What is Alibaba Cloud Realtime Compute for Apache Flink?

# 7.9. Synchronize data from a selfmanaged MySQL database hosted on ECS to an Elasticsearch cluster

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

# Prerequisites

- An Elasticsearch cluster of version 5.5, 5.6, 6.3, 6.7, or 7.4 is created. For more information, see Create an Elasticsearch cluster.
- The version of the self-managed MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.

# Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- DTS does not synchronize data definition language (DDL) operations. If a DDL operation is performed on a 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 cluster, 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

mapping of the table in the Elasticsearch cluster, perform DDL operations in the source MySQL database, and then pause and start the data synchronization task.

#### SQL operations that can be synchronized

INSERT, DELETE, and UPDATE

#### Data type conversion

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

## Before you begin

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

## Procedure

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

(?) Note On the buy page, set Source Instance to MySQL, set Target Instance to Elasticsearch, and set Synchronization Topology to One-Way Synchronization.

- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click **Data Synchronization**.
- 4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.
- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

#### Dat a Transmission Service

#### Data Synchronization-Synchronize d ata from a MySQL database to a dif ferent type of database

1.Configure Source and Destination	n 2.Select Objects to Synchronize	<u>&gt;</u>	3.Advanced Settings	$\rightarrow$	4.Precheck
Synchronization Task Name:	RDS_TO_Elasticsearch				
Source Instance Details					
Instance Type:	User-Created Database in ECS Instance	•			
Instance Region:	Singapore				
* ECS Instance ID:	the states	-			
Database Type:	MySQL				
	[				
* Port Number:	3306				
* Database Account:	dtstest				
* Database Password:	*****	<b>4</b> >			
Destination Instance Details					
Instance Type:	Elasticsearch				
Instance Region:	Singapore				
* Elasticsearch		_			
		Ť			
* Database Account:	elastic				
* Database Deservada		4			
Database Password:	*******	4>			

Set Whitelist and Next Cancel Parameter Section Description Synchroniz at DTS automatically generates a task name. We recommend that you N/A ion Task specify an informative name for easy identification. You do not need Name to use a unique task name. Instance Select User-Created Database in ECS Instance. Туре The source region that you selected on the buy page. You cannot Instance change the value of this parameter. Region ECS Instance Select the ID of the ECS instance that hosts the self-managed MySQL database. ID Database The value of this parameter is set to  $\ensuremath{\mathsf{MySQL}}$  and cannot be changed. Туре Port Number Enter the service port number of the self-managed MySQL database.

Source Instance

Details

Section	Parameter	Description		
	Dat abase Account	Enter the account of the self-managed MySQL database.		
		<b>Note</b> The database account must have the SELECT permission on the objects to be synchronized, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.		
	Database Password	Enter the password of the source database account.		
Destination Instance Details	Instance Type	The value of this parameter is set to <b>Elasticsearch</b> and cannot be changed.		
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.		
	Elasticsearch	Select the ID of the Elasticsearch cluster.		
	Dat abase Account	Enter the account that is used to connect to the Elasticsearch cluster. The default account is elastic.		
	Dat abase Password	Enter the password of the destination database account.		

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

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

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

1 Configure Source and Destination	2 Salact Objects to Synchroni	70	3 Advanced Settings		4 Drechack
	2.Select Objects to Synchron	20	5.Auvanceu Settings		4.FIGUICUK
Synchronization Mode: One-Way Synchronization					
Index Name: DatabseName_Tabl	Index Name: DatabseName_TableName				
Proccessing Mode In Existed Target Table:	check and Intercept $ \odot $ Ignore				
Available			Selected (To edit an object name Edit.) Learn more.	or its filter, hover ove	r the object and click
If you search globally, please ex	pand the I Q				
🕀 🪈 sys			dtstestdata (20biects)	ų	
🖃 🧧 dtstestdata			dtstestdata_customer Sour	rce Table Na	
		×	dtstestdata_order Source 1	Fable Na	
		>			
		<			
Select All	Select All		Select All		
*Name batch change:      Name batch change:     Source table DMS_ ONLINE_ Do y     want to copy the temporary table to     the target database during DDL:	o Ves ou Ves ONo ?				
* Retry Time for Failed Connection	720 Minutes	0			
				Cancel P	revious Next Precheck
Parameter	Description				
<ul> <li>Table Name         If you select Table Name, the name of the index that is created in the Elasticsearch cluster is the same as the name of the table. In this example, the index name is customer.     </li> <li>DatabaseName_TableName         If you select DatabaseName_TableName, the name of the index the created in the Elasticsearch cluster is <database name="">_<table dtstestdata_customer.<="" example,="" index="" is="" name="" p="" the="" this=""> </table></database></li> </ul>		s created in the ile. In this example, of the index that is _ <table name="">. In</table>			

Parameter	Description	
Processing Mode In Existed Target Table	<ul> <li>Pre-check and Intercept : checks whether the destination database contains indexes that have the same names as the source tables. If the destination database does not contain indexes that have the same names as the source tables, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started.</li> <li>Note If indexes in the destination database have the same names as the source tables, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Rename an object to be synchronized.</li> <li>Ignore: skips the precheck for identical index names in the source and destination databases.</li> <li>Ignore: skips the precheck for identical index names in the source and destination databases.</li> <li>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.</li> <li>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.</li> </ul>	
Select the objects to be synchronized	Select one or more objects from the <b>Available</b> section and click the > icon to move the objects to the <b>Selected</b> section. You can select tables or databases as the objects to be synchronized.	
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.	

Parameter	Description	
Paplicato Tomporany	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> </ul>	
Tables When DMS	data, the data synchronization task may be delayed.	
Operations	<ul> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> </ul>	
	Note If you select No, the tables in the destination database may be locked.	
Dotry Time for Epilod	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.	
Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.	

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

Edit Table	$\times$
Information: After corresponding table	r you edit the table or column name in the source database, the e or column name in the destination database is also updated.
* Index Name:	sys_config
* Type Name:	sys_config
Filter:	DTS supports the WHERE clause in SQL statements. Only data that meets the WHERE clause can be migrated to the destination
Settings_routing: (	Yes  No ⑦
_id value:	Bis id V Select an option.
Select All Colu	column mn Name Type param column param value
✓ set_by	varchar(12 index 🗸 false 🗸 add param
✓ set_time	timestamp index 🗸 false 🗸
✓ value	varchar(12 index 🗸 false 🗸 add param
✓ variable	varchar(12 index 🗸 false 🗸 add param
	ОК
Parameter	Description
Index Name	_
	For more information, see Terms.
Type Name	<ul> <li>Warning</li> <li>The only type of special characters that an index name and ty can contain is underscore ( ).</li> </ul>
	<ul> <li>To synchronize multiple source tables with the same schema destination object, you must repeat this step to set the same name and type name for the tables.</li> </ul>

Parameter	Description	
Filter	Specify SQL conditions to filter data. Only the data records that meet the specified conditions are synchronized to the destination cluster. For more information, see Use SQL conditions to filter data.	
lsPart it ion	Select whether to set partitions. If you select <b>Yes</b> , you must also specify the <b>partition key column</b> and <b>number of partitions</b> .	
Settings_routing	<ul> <li>Specify whether you want to store a document on a specified shard of the destination Elasticsearch cluster. For more information, see _routing.</li> <li>If you select Yes, you can specify custom columns for routing.</li> <li>If you select No, the _id value is used for routing.</li> <li>Note If the version of the destination Elasticsearch cluster is 7.4, you must select No.</li> </ul>	
_id value	<ul> <li>Primary key column Composite primary key fields are merged into one column.</li> <li>Business key If you select a business key, you must also specify the business key column.</li> </ul>	
	Select the <b>column parameter</b> and <b>parameter value</b> . For more information, see Mapping parameters in the Elasticsearch documentation.	
add param	<b>Note</b> DTS supports only the parameters that can be selected.	

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

## ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the 🕡 icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.
You can view the state of the data synchronization task on the Synchronization Tasks page.

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) <del>~</del>		Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1	> >>

## Check the index and data

If the data synchronization task is in the **Synchronizing** state, you can connect to the Elasticsearch cluster by using the Elasticsearch-Head plug-in. Then, you can check whether the index is created and data is synchronized as expected. For more information, see Use Cerebro to access an Elasticsearch cluster.

**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.

Elasticsearch http://e	.public.elasticse	es-	cn-		集群健康	ŧ值: g	reen (52 of 52)
the set of a set of the set of th							
Read .							
A 44 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	and a supervision of						
200	_index	_type	_id	_score ▲	address	id	name
.kibana_1	dtstestdata_customer	customer	2077	1		2077	Pati
.monitoring-es-6-2019.08.28	dtstestdata	customer	2079			2079	Harman Arstall
.monitoring-es-6-2019.08.29	dtstestdata_customer	customer	2083	1		2083	Katima Malaka
.monitoring-es-6-2019.08.30	dtstestdata_customer	customer	2087	1		2087	Electron
.monitoring-es-6-2019.08.31	dtstestdata_custoned	dex : dtste	stdata_o	customer",		2088	Sel
.monitoring-es-6-2019.09.01	dtstestdata_customend	": "2077",	2102	1		2102	Isis
.monitoring-es-6-2019.09.02	dtstestdata_custon_ve	rsion	2111	1		2111	Aar
.monitoring-es-6-2019.09.03	dtstestdata_customero	ore": 1, uting" <sup>[1</sup> 20	<b>-7</b> 416	1		2116	Rut
.monitoring-kibana-6-2019.08.28	dtstestdata custon.er	_source": {	2123	1		2123	Did
.monitoring-kibana-6-2019.08.29	dtstestdata customer	"address":	Ξ.,	1		2127	Sig
.monitoring-kibana-6-2019.08.30	dtstestdata customer	"id": 2077, "nomo": " D	al			2134	Ten
.monitoring-kibana-6-2019.08.31	dtstestdate custo }er	customer	2126			2126	location
.monitoring-kibana-6-2019.09.01	dtstestdata_ct_stomer	customer	2130	1		2130	Pok
.monitoring-kibana-6-2019.09.02	dtstestdata_customer	customer	2139	-		2139	Rec
.monitoring-kibana-6-2019.09.03	dtstestdata_customer		2157	1		2157	Bar
.security-6	dtstestdata_customer		2159	1		2159	Ald
dtstestdata_customer	dtstestdata_customer	customer	2165	1	and the second second	2165	Coc
dtstestdata_order	dtstestdata_customer	customer	2167	1		2167	Fiel
	dtstestdata_customer	customer	2168	1		2168	The
customer	dtstestdata_customer	customer	2180	1	-	2180	Ora
doc	dtstestdata_customer	customer	2185	1		2185	Gip
dtstestdata_order	dtstestdata_customer	customer	2187	1		2187	Erti

## 7.10. Synchronize data from an ApsaraDB RDS for MySQL instance to a MaxCompute project

MaxCompute (formerly 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 for MySQL instance to a MaxCompute project by using Data Transmission Service (DTS).

## Prerequisites

The following operations are performed:

- Activate MaxCompute and DataWorks
- Create a MaxCompute project

#### Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- Only tables can be selected as the objects to 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 PRIMARY KEY constraints. 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:

- Self-managed database that is hosted on Elastic Compute Service (ECS)
- Self-managed database that is connected over Express Connect, VPN Gateway, or Smart Access Gateway
- Self-managed database that is connected over Database Gateway
- ApsaraDB RDS for MySQL instance that is owned by the same Alibaba Cloud account as the MaxCompute project or a different Alibaba Cloud account from the MaxCompute project

This topic uses an **ApsaraDB RDS for 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 self-managed MySQL database, you must deploy the network environment for the source database. For more information, see **Preparation overview**.

## SQL operations that can be synchronized

• DDL operation: ADD COLUMN

(?) Note Only the following data types are supported: INTEGER, BIGINTEGER, IGINT, LONGSTRING, YEAR, TIME, DATA, TIME STAMP, DATA TIME, BYTE, BOOLEAN, DECIMAL, DOUBLE, and FLOAT.

• 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, for example, customer\_log. Then, DTS synchronizes the incremental data that was generated in the source database to the incremental data table.

⑦ Note For more information, see Schema of an incremental data table.

## Procedure

**Warning** To ensure that the synchronization account can be authorized, we recommend that you perform the following steps by using your Alibaba Cloud account.

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

On the buy page, set Source Instance to MySQL, set Destination 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.

#### Dat a Transmission Service

#### Data Synchronization Synchronize d ata from a MySQL database to a dif ferent type of database

1.Configure Source and Destinatio	n 2.Authorize MaxCompute Account		3.Select Objects to Synchronize	4.Precheck
Synchronization Task Name:	RDS_TO_MaxCompute			
Source Instance Details				
Instance Type:	RDS Instance	•		
Instance Region:	Singapore			
* Instance ID:	rm-	-	RDS Instances of Other Apsara Stack Accounts	
* Database Account:	dtstestaccount			
* Database Password:	•••••	4>		
Destination Instance Dataile				
Destination Instance Details				
Instance Type:	MaxCompute			
Instance Decian	Cinconere			
instance Region.	Singapore			
* Project:	dtstest			
				Cancel Set Whitelist and Next

Section	Parameter	Description			
N/A	Synchronizatio n Task Name	DTS automatically generates a task name. We recommend that you specify an informative name to identify the task. You do not need to use a unique task name.			
	Instance Type	Select RDS Instance.			
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.			
	Instance ID	Select the ID of the source RDS instance.			
		Enter the database account of the source RDS instance.			
	Dat abase Account	<b>Note</b> 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.			
	Dat abase Password	Enter the password of the database account.			
Source					
Instance Details					

Section	Parameter	Description			
		Select <b>Non-encrypted</b> or <b>SSL-encrypted</b> . If you want to select <b>SSL-encrypted</b> , 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 ApsaraDB RDS for MySQL instance.			
	Encryption	<b>Notice</b> The <b>Encryption</b> parameter is available only for regions in the Chinese mainland and the China (Hong Kong) region.			
	Instance Type	This parameter is set to <b>MaxCompute</b> and cannot be changed.			
Destination	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.			
Instance Details	Project	Enter the name of the MaxCompute project. You can search for a project on the Workspaces page in the DataWorks console.			

#### 7.

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.

1.Configure Sour	ce and Destination	2.Authorize MaxCompute Account		3.Select Objects to Synchronize	>	4.Precheck	
-	To synchronize data to a MaxCom	pute instance, you must grant the following per	rmissions of pr	oject dtstest to the synchronization accou	nt.		
	CreateTable						
	CreateInstance						
	CreateResource						
	CreateJob						
	List						
						Cancel Previo	us Next

9. Select the synchronization policy and the objects to be synchronized.

Select	Partition Name	Field Type	Description	
<b>v</b>	modifytime_year	String	Year of Incremen	tal Update
<b>&gt;</b>	modifytime_month	String	Month of Increme	antal Update
<b>&gt;</b>	modifytime_day	String	Date of Incremen	ital Update
<b>~</b>	modifytime_hour	String	Hour of Incremer	ntal Update
	modifytime_minute	String	Minute of Increm 15 minutes.)	ental Update (Incremental data is written into a separate partition every
Note: do n cleans up t roccessing N xisted Targe Available	ot clean up the incremental data log the log too early, the DTS incrementa Mode In it Table:	generated by the source il task may fail rrcept O Ignore	database after the D	TTS task is started when the DTS full task is running. If the source database Selected (To edit an object name or its filter, hover over the object and cli Edit.) Learn more.
Expand the	ne tree before you perform a glol ables 02 _polar2	Q	> <	Q dtstest123 Source Database Name (10bjects) ≣ tvv02
Select All				Select All

Setting	Description
Partition Definition of Incremental Data Table	Select the partition names based on your business requirements. For more information, 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 and then synchronizes incremental data.

Setting	Description
	• <b>Pre-check and Intercept</b> : 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.
	<b>Note</b> You can use the object name mapping feature to rename the tables that are synchronized to the destination database. You can use this feature if the source and destination databases contain identical table names and the tables in the destination database cannot be deleted or renamed. For more information, see Rename an object to be synchronized.
Select the processing mode of conflicting	<ul> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> </ul>
tables	<ul> <li>Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks.</li> <li>During initial data synchronization, DTS does not synchronize the data records that have the same primary keys as the data records in the destination database. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization.</li> <li>If the source and destination databases have different schemas, initial data synchronized or the data synchronization task fails.</li> </ul>
	Select one or more tables from the <b>Available</b> section and click the > icon
	to move the tables to the <b>Selected</b> section.
Select the objects to be synchronized	<ul> <li>Note</li> <li>You can select tables from multiple databases as the objects to be synchronized.</li> <li>By default, after an object is synchronized to the destination instance, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Rename an object to be synchronized.</li> </ul>

Setting	Description
Whether to enable the	After DTS synchronizes data to MaxCompute, DTS adds additional columns to the destination table. If the names of additional columns are the same as the names of existing columns in the destination table, data synchronization fails. Select <b>Yes</b> or <b>No</b> to specify <b>whether you want to enable the new naming rules for additional columns</b> .
new naming rules for additional columns	<b>Warning</b> Before you specify this parameter, check whether additional columns and existing columns in the destination table have name conflicts. For more information, see Naming rules for additional columns.
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.
	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> </ul>
Replicate Temporary Tables When DMS Performs DDL Operations	<ul> <li>Note If online DDL operations generate a large amount of data, the data synchronization task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by</li> </ul>
	online DDL operations. Only the original DDL data of the source database is synchronized.
	<b>Note</b> If you select No, the tables in the destination database may be locked.
Retry Time for Failed	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the Synchronizing state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1 >	> >>

## Schema of an incremental data table

Note You must run the set odps.sql.allow.fullscan=true; command in MaxCompute to allow full table scan for the MaxCompute project.

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.

Field

Description

Field	Description
record_id	The ID of the incremental log entry.
	<ul> <li>Note</li> <li>The ID auto-increments for each new log entry.</li> <li>If an UPDATE operation is performed, DTS generates two incremental log entries to record the pre-update and post-update values. The two incremental log entries have the same record ID.</li> </ul>
operation_flag	<ul> <li>The operation type. Valid values:</li> <li>I: an INSERT operation</li> <li>D: a DELETE operation</li> <li>U: an UPDATE operation</li> </ul>
utc_timestamp	The operation timestamp, in UTC. It is also the timestamp of the binary log file.
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

The **before\_flag** and **after\_flag** fields of an incremental log entry are defined depending on the operation type.

• INSERT

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



• UPDATE

DTS generates two incremental log entries for an UPDATE operation. The two incremental log entries have the same values for the record\_id, operation\_flag, and utc\_timestamp fields.

The first 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.



DELET E

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.

		В			E							
	id 🗸	register_time 🗸 🗸	address 🗸 🗸	record_id 🗸 🗸	operation_flag 🍸	utc_timestamp 🗸	before_flag 🗸	after_flag 🗸	modifytime_year 🗸	modifytime_month 🗸	modifytime_day 🗸	modifytime_hour 🗸
I.	9999	2016-11-18 11:44:54		156	D	156 845	Y	N	2019	08	16	16

## Merge a full baseline table and an 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 a table named customer. The following figure shows the schema of the customer table.

	Field	Туре	Ŧ	Null	Ŧ	Кеу	Ŧ	Default	-	Extra	Ŧ	
1	id	int(11)		NO		PRI						
2	register_time	timestamp		YES								
3	address	varchar(32)		YES								

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 the 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 <coll>,
      <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, af
ter flag, <coll>, <col2>, <colN>
 from(
select incr.record id, incr.operation flag, incr.after flag, incr.<coll>, incr.<col2>,i
ncr.<colN>
 from  incr
where utc timestamp< <timestamp>
union all
select 0 as record id, 'I' as operation flag, 'Y' as after flag, base.<coll>, base.<col
2>,base.<colN>
 from  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, af
ter_flag, id, register_time, address
 from(
select incr.record_id, incr.operation_flag, incr.after_flag, incr.id, incr.register_tim
e, 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.



# 7.11. Synchronize data from an ApsaraDB RDS for MySQL instance to a Tablestore instance

Tablestore is a multi-model data storage service that is developed by Alibaba Cloud. Tablestore can store large volumes of structured data by using multiple models, and supports fast data query and analytics. This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to a Tablestore instance by using Data Transmission Service (DTS). You can also follow the procedure to synchronize data from a self-managed MySQL database to a Tablestore instance. The data synchronization feature allows you to transfer and analyze data with ease.

## Prerequisites

A Tablestore instance is created. For more information, see Create instances.

## Precautions

- During initial full data synchronization, DTS consumes the read and write resources of the source and destination databases. This may increase the loads of the database servers. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination databases. We recommend that you synchronize data during off-peak hours.
- DTS does not synchronize data definition language (DDL) operations. If a DDL operation is performed on a table in the source database during data synchronization, you must perform the following steps: Remove the table from the selected objects, remove the table from the Tablestore instance, and then add the table to the selected objects again. For more information see Remove an object from a data synchronization task and Add an object to a data synchronization task.
- You can synchronize at most 64 tables to the Tablestore instance. If you need to synchronize more than 64 tables to the Tablestore instance, submit a ticket to raise the limit.
- The names of the tables and columns to be synchronized must comply with the naming conventions of the Tablestore instance.
  - The name of a table or an index can contain letters, digits, and underscores (\_). The name must start with a letter or underscore (\_).
  - The name of a table or an index must be 1 to 255 characters in length.

## Initial synchronization types

Initial synchronization type	Description
	DTS synchronizes the schemas of tables from the source database to the destination database.
Initial schema synchronization	• Warning MySQL and Tablestore are heterogeneous databases. DTS does not ensure that the schemas of the source and destination databases are consistent after initial schema synchronization. We recommend that you evaluate the impact of data type conversion on your business. For more information, see Data type mappings for schema synchronization.

Initial synchronization type	Description
Initial full data synchronization	DTS synchronizes historical data of tables from the source database to the destination database. Historical data is the basis for subsequent incremental data synchronization.
Initial incremental data synchronization	DTS synchronizes incremental data from the source database to the destination database in real time. The following SQL operations can be synchronized during initial incremental data synchronization: INSERT, UPDATE, and DELETE.
	<b>Warning</b> We recommend that you do not execute DDL statements in the source database. Otherwise, data synchronization may fail.

## Before you begin

**Notice** When you configure the destination database, you must specify the AccessKey pair. To protect the AccessKey pair of your Alibaba Cloud account, we recommend that you grant permissions to a RAM user and create an AccessKey pair for the RAM user.

#### 1. Create a RAM user and grant the AliyunOTSFullAccess permission on Tablestore to the RAM user.

- i. Log on to the RAM console.
- ii. Create a RAM user.
- iii. In the left-side navigation pane, choose **Identities > Users**.
- iv. On the Users page, find the RAM user and click Add Permissions in the Actions column.

RAM		RAM / Users	
Overview		Users	
Identities	^	A RAM user is an identity entity. It represents a user or application in your organization that needs to access cloud resources.	
Groups		You can manage users in the following steps:	
Users		1.Create a RAM user, and set a password for this user to log on to the console or create an AccessKey for the application to call APIs. 2.Add the user to a group. To perform this operation, you must have created a group and granted permissions to it.	
Settings			
SSO		Create User User Logon Name V Enter Q	
Permissions	~	User Logon Name/Display Name Note Created Actions	
Grants		dtstest@ Sep 19, 2019, 14:06:10 Add to Group Add Permissions Delete dtstest	
Policies			
RAM Roles			

v. In the dialog box that appears, enter *AliyunOTSFullAccess* in the search box, and click the policy name to add the policy to the Selected section.

You can add a maximum of 5 policies. To add more policies, repeat the operation.   Authorization   Alibaba Cloud account all resources   Specified Resource Group   Enter a resource group name.   Principal   dtstest@cores.onaliyun.com ×   Select Policy   System Policy   Custom Policy   Y Create Policy   System Policy   Custom Policy   Authorization Policy Name   Description   AliyunOTSFullAccess   Provides full access to Table Store(OTS) via Management	Add Permissions			
Authorization  Alibaba Cloud account all resources  Specified Resource Group  Enter a resource group name.  Principal  dtstest@cores.onaliyun.com ×  Select Policy  System Policy Custom Policy + Create Policy  Select Policy  Selected (1) Clear  AliyunOTSFullAccess  Authorization Policy Name Description  AliyunOTSFullAccess Provides full access to Table Store(OTS) via Management  2	You can add a maximum	of 5 policies. To add more policies, repeat the operation.		
Alibaba Cloud account all resources Specified Resource Group Enter a resource group name. Principal dtstest@cores.onaliyun.com × Select Policy System Policy Custom Policy + Create Policy Selected (1) Clear AliyunOTSFullAccess 1 AliyunOTSFullAccess × Authorization Policy Name Description AliyunOTSFullAccess Provides full access to Table Store(OTS) via Management	Authorization			
Specified Resource Group   Enter a resource group name.     Principal   dtstest@cores.onaliyun.com ×   Select Policy   System Policy   Custom Policy   + Create Policy   Selected (1)   AliyunOTSFullAccess   Authorization Policy Name   Description   AliyunOTSFullAccess   Provides full access to Table Store(OTS) via Management	) Alibaba Cloud account all resou	irces		
Enter a resource group name.  Principal  dtstest@cores.onaliyun.com X  Select Policy  System Policy Custom Policy + Create Policy  AliyunOTSFullAccess  Authorization Policy Name Description  AliyunOTSFullAccess Provides full access to Table Store(OTS) via Management  2	) Specified Resource Group			
Principal dtstest@cores.onaliyun.com X Select Policy System Policy Custom Policy + Create Policy AliyunOTSFullAccess 1 Selected (1) Clear AliyunOTSFullAccess 2 AliyunOTSFullAccess X Authorization Policy Name Description AliyunOTSFullAccess Provides full access to Table Store(OTS) via Management 2	Enter a resource group name.			$\sim$
System Policy       Custom Policy       + Create Policy       Selected (1)       Clear         AliyunOTSFullAccess       1       S       AliyunOTSFullAccess       X         Authorization Policy Name       Description       2       2         AliyunOTSFullAccess       Provides full access to Table Store(OTS) via Management       2	Principal dtstest@cores.onaliyun.com Select Policy	×		
AliyunOTSFullAccess     1     S     AliyunOTSFullAccess     X       Authorization Policy Name     Description     2     2	System Policy Custom Polic	ty + Create Policy	Selected (1)	Clear
Authorization Policy Name     Description       AliyunOTSFullAccess     Provides full access to Table Store(OTS) via Management	AliyunOTSFullAccess 1	8	AliyunOTSFullAccess	×
AliyunOTSFullAccess Provides full access to Table Store(OTS) via Management	Authorization Policy Name	Description		
	AliyunOTSFullAccess	Provides full access to Table Store(OTS) via Management	2	

- vi. Click OK.
- vii. Click Complete.
- 2. Create an AccessKey pair for the RAM user. For more information, see Create an AccessKey pair.
- 3. (Optional) If you need to configure a data synchronization task as a RAM user, you must grant the AliyunDT SDefault Role permission to the RAM user. For more information, see Authorize DTS to access Alibaba Cloud resources.

**?** Note If you use an Alibaba Cloud account to configure a data synchronization task, skip this step.

#### Procedure

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

**?** Note On the buy page, set Source Instance to MySQL, set Destination Instance to Tablestore, and set Synchronization Topology to One-way Synchronization.

2.

- 3.
- 4.
- 5.
- 6. Configure the source and destination databases.

#### Dat a Transmission Service

#### Data Synchronization Synchronize d ata from a MySQL database to a dif ferent type of database

1 Select Source and Des	tination	2 Select Obi	iect to Be Synchronized	3.A	dvanced Settings 4 Precheck		
Synchronization T	ask Name:	MySQL_To_TableStore					
Source Instance Details							
Insta	ance Type:	RDS Instance		~			
Instan	ce Region:	China (Hangzhou)					
* In	stance ID:	rm-bp		•	RDS Instances of Other Apsara Stack Accounts		
* Databas	e Account:	dtstest					
* Database	Password:						
* E	Encryption:	Non-encrypted Os	SL-encrypted				
Destination Instance Details							
Insta	ance Type:	Tablestore		~			
Instan	ce Region:	China (Shanghai)					
* In	istance ID:	table-dtstest		•			
* Acce	essKeyId:	LTA					
* AccessKe	vSecrect:		*****				
					Cancel Set Whitelist and Next		
Section	Para	ameter	Description				
N/A	Syno n T a	chronizatio ask Name	DTS automatically gen you specify an informa need to use a unique t	iera ativ asl	ates a task name. We recommend that re name to identify the task. You do not k name.		
	Inst	ance Type	Select RDS Instance.				
	Inst Reg	ance ion	The source region that you selected on the buy page. You cannot change the value of this parameter.				
	Inst	ance ID	Select the ID of the source RDS instance.				
	Dat abase Account		Enter the database account of the source RDS instance. The account must have the SELECT permission on the objects to be synchronized, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.           ⑦         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.				
Source Instance Details	Data Pass	abase sword	Enter the password of	th	e database account.		

Section	Parameter	Description			
	Encryption	Select <b>Non-encrypted</b> or <b>SSL-encrypted</b> . If you want to select <b>SSL-encrypted</b> , 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 ApsaraDB RDS for MySQL instance.			
		<b>Notice</b> The <b>Encryption</b> parameter is available only for regions in the Chinese mainland and the China (Hong Kong) region.			
	Instance Type	Select Tablestore.			
	Instance	The destination region that you selected on the buy page. You			
	Region	cannot change the value of this parameter.			
Destination Instance	Region Instance ID	cannot change the value of this parameter. Select the ID of the destination Tablestore instance.			
Destination Instance Details	Region Instance ID AccessKeyld	cannot change the value of this parameter. Select the ID of the destination Tablestore instance. Enter the AccessKey ID. For more information, see Create an AccessKey pair.			

7.

- 8. Configure the synchronization policy and the objects to be synchronized.
  - i. Configure the synchronization policy.

1.Select Source and Desti	ination 2.Select Object	t to Be 3.A	dvanced Settings	> 4.Precheck
Initial Synchronization:	✓ Initial Schema Synchronization	✓ Initial Full Data Synchroniza	tion 🔽 Initial Increme	ental Data Synchronization
Proccessing Mode In Existed Target Table:	Pre-check and Intercept $ \bigcirc $ Ignore			
Merge Multi Tables:C	) Yes 💿 No			
Synchronization Type:	🗹 Insert 🔽 Update 🔽 D	elete		
Dirty Row Method:	Skip 🗸			
Update Mode:	Line overlay 🗸			
Batch Request Type:	BulkImportRequest 🗸			
Queue Size:	1024	Collapse		
Thread Count:	8			
Concurrency:	8			
Bucket Count:	2			

Parameter	Description				
Initial Synchronization	Select Initial Schema Synchronization, Initial Full Data Synchronization, and Initial Incremental Data Synchronization. For more information, see Initial synchronization types.				
	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.				
	<b>Note</b> You can use the object name mapping feature to rename the tables that are synchronized to the destination database. You can use this feature if the source and destination databases contain identical table names and the tables in the destination database cannot be deleted or renamed. For more information, see Rename an object to be synchronized.				
	<ul> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> </ul>				
Processing Mode In Existed Destination Table	<b>Warning</b> If you select <b>Ignore</b> , data consistency is not guaranteed and your business may be exposed to potential risks.				
	<ul> <li>During initial data synchronization, DTS does not synchronize the data records that have the same primary keys as the data records in the destination database. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization.</li> </ul>				
	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		
Merge Multi Tables	<ul> <li>Yes: In online transaction processing (OLTP) scenarios, sharding is implemented to speed up the response to business tables. You can merge multiple source tables that have the same schema into a single destination table. This feature allows you to synchronize data from multiple tables in the source database to a single table in the Tablestore instance.</li> <li>Note <ul> <li>DTS adds a column nameddts_data_source to the destination table in the Tablestore instance. This column is used to record the data source. The data type of this column is varchar. DTS specifies the column values based on the following format: <data :<source="" database="" idd="" instance="" name="" synchronization="">.Source table name&gt; .Such column values allow DTS to identify each source table. For example, dts*******:dtstestdata.customer1</data></li> <li>If you set this parameter to Yes, all of the selected source tables in the task are merged into the destination table. If you do not need to merge specific source tables, you can create a separate data synchronization task for these tables.</li> </ul> </li> <li>No is selected by default.</li> </ul>		
Synchronization Type	Select the types of operations that you want to synchronize based on your business requirements. All operation types are selected by default.		
Processing Policy of Dirty Data	<ul> <li>Select a processing policy for handling data write errors:</li> <li>Skip</li> <li>Block</li> </ul>		

Parameter	Description
Data Write Mode	<ul> <li>Update Row: The PutRowChange operation is performed to update rows in the Tablestore instance.</li> <li>Overwrite Row: The UpdateRowChange operation is performed to overwrite rows in the Tablestore instance.</li> </ul>
Batch Write Operation	<ul> <li>Select the operation used to write multiple rows of data to the Tablestore instance.</li> <li>BulkImportRequest: Data is written offline.</li> <li>BatchWriteRowRequest: Data is written in batches.</li> <li>To achieve higher read and write efficiency and reduce your costs of using the Tablestore instance, we recommend that you select</li> <li>BulkImportRequest.</li> </ul>
More	
└ Queue Size	Enter the length of the queue for writing data to the Tablestore instance.
└─ T hread Quantity	Enter the number of callback threads for writing data to the Tablestore instance.
└ Concurrency	Enter the maximum number of concurrent threads for the Tablestore instance.
└ Buckets	Enter the number of concurrent buckets for incremental and sequential writes. To improve the concurrent write capability, you can set this parameter to a relatively larger value.
	<b>Note</b> The value must be less than or equal to the maximum number of concurrent threads.

ii. Select the objects to be synchronized.

Available		Selected (To edit an object name or its filter, he	over over the object and click	
Expand the tree before you perform a i i i online i i mysql_xtdkamka i i custm_info i i sdc i dtstestdata i i tables ii child ii customer0421 ii neworder ii parent ii tableddl iii test_customer i i odts i o		Edit.) Learn more.	Q 20bjects)	
*Change Mapped Name:	Do Not Change Database and Table Names	Change Database and Table Names		
			Cancel Previous N	lext
Setting	Description			
Select the objects to be synchronized	Select one or more tables f to move the tables to the S selected. By default, after an object is name of the object remains mapping feature to rename destination instance. For mo synchronized.	rom the <b>Available</b> section <b>Gelected</b> section. Up to 64 is synchronized to the desting unchanged. You can use the the objects that are synch- ore information, see Renam	and click the > icon tables can be nation instance, the ne object name ronized to the e an object to be	n
	• warning If you set select multiple source tak feature to change their na instance. Otherwise, data	t <b>Merge Multi Tables</b> to <b>Y</b> bles, you must use the obje ames to the same table nar a is inconsistent.	' <b>es in <mark>Step 8</mark> and</b> ct name mapping me in the Tablestore	

#### iii. Click Next .

9. (Optional)In the **Selected** section, move the pointer over a table, and then click **Edit**. In the Edit Table dialog box, set the data type of each column in the Tablestore instance.

nformation: corresponding	After you edit the table or column r table or column name in the destina	name in the sou ation database is	rce database, s also updated	the
* Table Name	customer			
Filte	DTS supports the WHERE claus statements. Only data that mee clause can be migrated to the c	e in SQL ets the WHERE destination	• •	erify
DML and I Staten Filter	DDL ient ing Select DDL or DML stateme	ents. 🕧		
DML F	lter: 🗹 insert 🔽 update 🔽 dele	ete		
DML Fi	lter: 🗹 insert 🗹 update 🗹 dele Column Name	ete Type	primaryKey	targetType
DML Fi	Iter: 🗹 insert 🔽 update 🗹 dele Column Name address	Type varchar(32)	primaryKey	targetType String
DML F	Iter:  insert  update  dele Column Name address id	ete Type varchar(32) int(11)	primaryKey O	targetType String ✓ Integer Binary String Double Boolean

10. Configure the primary key columns of the tables that you want to synchronize to the Tablestore instance.

1.Select Source and Destination	2.Select Object to Be Synchronized	3.Advanced Settings	4.Precheck
Tablestore Table Group	Tablestore Table Name	Primary Key Column	Definition Status(All) 👻
table-dtstest	customer	id	Defined
table-dtstest	order	orderid	Defined
Enter an Tablestore table name. Search		Total: 2 item(s	s), Per Page: 20 $\checkmark$ item(s) $\ll$ $\langle$ 1 $\rangle$ »
			Cancel Previous Save Precheck
? Note For more in	nformation about prima	ry keys of Tablestore	instances, see Primary keys.

- 11.
- 12.
- 13.

## 7.12. Synchronize data from a selfmanaged MySQL database to a Message Queue for Apache Kafka instance

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

## Prerequisites

- The version of the self-managed MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The version of the destination Message Queue for Apache Kafka instance is 0.10.1.0 to 2.x.
- In the destination Message Queue for Apache Kafka instance, a topic is created to receive the synchronized data. For more information, see Create a topic.

#### Context

Message Queue for Apache Kafka is a distributed, high-throughput, and scalable message queue service that is provided by Alibaba Cloud. It provides fully managed services for the open source Apache Kafka to resolve the long-standing short comings of open source products. Message Queue for Apache Kafka allows you to focus on business development without spending much time in deployment and O&M. Message Queue for Apache Kafka is used in big data scenarios such as log collection, monitoring data aggregation, streaming data processing, and online and offline analysis. It is important for the big data ecosystem.

## Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.

## Limits

- Only tables can be selected as the objects to synchronize.
- DTS does not synchronize the data in a renamed table to the destination Kafka cluster. This applies if the new table name is not included in the objects to synchronize. If you want to synchronize the data in a renamed table to the destination Kafka cluster, you must **reselect the objects to be synchronized**. For more information, see Add an object to a data synchronization task.

## Before you begin

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

#### Procedure

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

**?** Note On the buy page, set Source Instance to MySQL, set Dstination Instance to Kafka, and set Synchronization Topology to One-way Synchronization.

- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.
- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

Synchronization Task Name:	MySQL_TO_Kafkaold			
Source Instance Details				
Instance Type:	User-Created Database in EC	S Instance 🗸		
Instance Region:	China (Hangzhou)			
* ECS Instance ID:	i-	•		
Database Type:	MySQL			
* Port Number:	3306			
* Database Account:	dtstest			
* Database Password:	******	4>		
Destination Instance Details				
Instance Type:	User-Created Database Conn	ected Over Express Conne 🗸		
Instance Region:	China (Hangzhou)			
* Peer VPC:	vpc-	-		
Database Type:	Kafka	~		
* IP Address:	172.16			
* Port Number:	9092		1	
Database Account:			Optional	
Database Password:		4>	Optional	
* Topic:	dtstesttopic	~	Get Toplic list	
	Click Get Topic List and then s	elect the specific topic.		
* Kafka Version	0.10	~		
* Encryption:	Non-encrypted O SCRAM-S	GHA-256		
				Cancel Set Whitelist and Next
Section	Parameter	Description		
None	Synchronizat ion Task	DTS automatic specify an info	cally generates a task name. We prmative name to identify the ta	recommend that you sk. You do not need

to use a unique task name.

Name

Section	Parameter	Description
	lnstance Type	Select an instance type based on the deployment of the source database. In this example, select <b>User-Created Database in ECS</b> <b>Instance</b> . You can also follow the procedure to configure data synchronization tasks for other types of self-managed MySQL databases.
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.
	ECS Instance ID	Select the ID of the Elastic Compute Service (ECS) instance that hosts the self-managed MySQL database.
Source Instance Details	Database Type	This parameter is set to <b>MySQL</b> and cannot be changed.
	Port Number	Enter the service port number of the self-managed MySQL database.
	Database Account	Enter the account of the self-managed MySQL database. The account must have the SELECT permission on the required objects, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.
	Dat abase Password	Enter the password of the database account.
		Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway.
	lnstance Type	<b>Note</b> You cannot select Message Queue for Apache Kafka as the instance type. You can use Message Queue for Apache Kafka as a self-managed Kafka database to configure data synchronization.
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.

Section	Parameter	Description		
Destination Instance Details	Peer VPC	Select the ID of the virtual private cloud (VPC) to which the destination Message Queue for Apache Kafka instance belongs. To obtain the VPC ID, you can log on to the Message Queue for Apache Kafka console and go to the Instance Details page of the Message Queue for Apache Kafka instance. In the <b>Basic Information</b> section, you can view the VPC ID. Instance Details Task Records Basic Information Instance ID: Instance Type: Standard Edition Peak Traffic: 20 MB/s Disk Type: Ultra Disk VPC ID: vpc-bi		
	Database Type	Select <b>Kafka</b> .		
	IP Address	Enter an IP address that is included in the <b>Default Endpoint</b> parameter of the Message Queue for Apache Kafka instance. <b>Note</b> To obtain an IP address, you can log on to the Message Queue for Apache Kafka console and go to the Instance Details page of the Message Queue for Apache Kafka instance. In the <b>Basic Information</b> section, you can obtain an IP address from the <b>Default Endpoint</b> parameter.		
	Port Number	Enter the service port number of the Message Queue for Apache Kafka instance. The default port number is 9092.		
	Database Account	Enter the username that is used to log on to the Message Queue for Apache Kafka instance. <b>Note</b> If the instance type of the Message Queue for Apache Kafka instance is VPC Instance, you do not need to specify the database account or database password.		
	Dat abase Password	Enter the password of the username.		
	Торіс	Click <b>Get Topic List</b> and select a topic name from the drop-down list.		

Section	Parameter	Description
	Kafka Version	Select the version of the Message Queue for Apache Kafka instance.
	Encryption	Select <b>Non-encrypted</b> or <b>SCRAM-SHA-256</b> based on your business and security requirements.

#### 7.

8. Select the objects to synchronize.

1.Select Source and Destination	$\rightarrow$	>	4.Precheck	
Synchronization Mode:One-Way Synchroniz Data format delivered to Kafka:  DTS Avro C Synchronize to Kafka Partition Policy: Post all to Partitio O Delivered differer Note: Affer the data	ation anal Json ⑦ t parts by hash value of library name + table t parts by hash value of primary key synchronization operation is officially started,	name do not modify the number of partitions of the	target topic, otherwise the	
Available Expand the tree before you perform a glo Christian Christ		Selected (To edit an object name or its filte Edit.) Learn more.	er, hover over the object and click	
Select All		Select All		
<ul> <li>*Rename Databases and Tables:</li> <li>* Retry Time for Failed Connection</li> </ul>	Do Not Change Database and Table Names 20 Minutes 🕐	Change Database and Table Names		
			Cancel Previous	Next

Parameter	Description
Data Format in Kafka	The data that is synchronized to the Kafka cluster is stored in the Avro or Canal JSON format. For more information, see Data formats of a Kafka cluster.
Policy for Shipping Data to Kafka Partitions	The policy used to synchronize data to Kafka partitions. Select a policy based on your business requirements. For more information, see Specify the policy for synchronizing data to Kafka partitions.

Parameter	Description	
	Select one or more tables from the <b>Available</b> section and click the <b>&gt;</b> icon to add the tables to the <b>Selected</b> section.	
Select the objects to synchronize	<b>Note</b> DTS maps the table names to the topic name that you select in Step 6. If you want to rename the topic, you can use the object name mapping feature. For more information, see <b>Rename an object to be synchronized</b> .	
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.	
Retry Time for Failed	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.	
Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.	

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

#### 10. Configure initial synchronization.

Initial Synchronization: 🗹 Initial Schema Synchronization 🛛 Initial Full Data Synchronization Note: Trigger synchronization is not supported, please Reference Document Filter options: 🔽 Ignore DDL in incremental synchronization phase				
		Cancel	Previous Save Precheck	
Setting	Description			
Initial Synchronization	Select both <b>Initial Schema Sync</b> <b>Synchronization</b> . DTS synchroniz required objects and then synchro	hronization and Init es the schemas and h nizes incremental dat	<b>ial Full Data</b> historical data of the a.	
Filter options	<b>Ignore DDL in incremental syn</b> default. In this case, DTS does no performed on the source databas	chronization phase synchronize DDL ope during incremental	is selected by erations that are data synchronization.	

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.
- 12. Close the **Precheck** dialog box after the following message is displayed: **Precheck Passed**. Then, the data synchronization task starts.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Act	tions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task   Switc Subscription   Upgr N	:h to rade 4ore
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < <u>1</u> >	»

## 7.13. Synchronize data from an ApsaraDB RDS for MySQL instance to a self-managed Kafka cluster

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

#### Prerequisites

- A Kafka cluster is created and the Kafka version is 0.10.1.0 to 2.7.0.
- An ApsaraDB RDS for MySQL instance is created. For more information, see Create an ApsaraDB RDS for MySQL instance.

#### Precautions

 DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.

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

## Limits

- Only tables can be selected as the objects to synchronize.
- DTS does not synchronize the data in a renamed table to the destination Kafka cluster. This applies if the new table name is not included in the objects to synchronize. If you want to synchronize the data in a renamed table to the destination Kafka cluster, you must **reselect the objects to be synchronized**. For more information, see Add an object to a data synchronization task.

## Supported synchronization topologies

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

## Procedure

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

**Note** On the buy page, set Source Instance to MySQL, Destination Instance to Kafka, and Synchronization Topology to One-Way Synchronization.

- 2.
- 3.
- 4.
- 5.
- 6. Configure the source and destination instances.

Cancel Set Whitelist and Next

Synchronization Task Name:	MySQL_TO_Kafka	
Source Instance Details		
Instance Type:	RDS Instance 🗸	
Instance Region:	China (Hangzhou)	
* Instance ID:	rm-1	RDS Instances of Other Apsara Stack Accounts
* Database Account:	dtstest	
* Database Password:	••••••••••••••••••••••••••••••••••••••	
* Encryption:	Non-encrypted     O SSL-encrypted	
Destination Instance Details		
Instance Type:	User-Created Database in ECS Instance 🗸	
Instance Region:	China (Hangzhou)	
* ECS Instance ID:	-	
Database Type:	Kafka 🗸	
* Port Number:	9092	
Database Account:		Optional
Database Password:	٩>	Optional
* Topic:	rdstest 🗸	Get Toplic list
	Click Get Topic List and then select the specific topic.	
* Kafka Version	0.10 ~	
* Encryption:	Non-encrypted O SCRAM-SHA-256	

Section	Parameter	Description
None	Synchroniz <i>a</i> t ion Task Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.
	lnstance Type	Select RDS Instance.
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.
	Instance ID	The ID of the source ApsaraDB RDS for MySQL instance.
	Database Account	The account that is used to connect to the source database. The account must have the SELECT permission on the required objects and the REPLICATION CLIENT, REPLICATION SLAVE, and SHOW VIEW permissions.
	Database Password	The password of the source database account.
Source Instance Details		

> Document Version: 20220712

Section	Parameter	Description
	Encryption	Select <b>Non-encrypted</b> or <b>SSL-encrypted</b> . If you select <b>SSL-encrypted</b> , you must enable SSL encryption for the ApsaraDB RDS for MySQL instance before you configure the data synchronization task. For more information, see Configure SSL encryption for an ApsaraDB RDS for MySQL instance.
		Notice The Encryption parameter is available only for regions in the Chinese mainland and the China (Hong Kong) region.
		The instance type of the Kafka cluster. In this example, <b>User</b> - <b>Created Database in ECS Instance</b> is selected.
	lnstance Type	<b>Note</b> If you select other instance types, you must deploy the network environment for the Kafka cluster. For more information, see <b>Preparation overview</b> .
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.
	ECS Instance ID	The ID of the Elastic Compute Service (ECS) instance on which the Kafka cluster is deployed.
	Database Type	Select <b>Kafka</b> .
	Port Number	The service port number of the Kafka cluster. Default value: 9092.
	Dat abase Account	The username that is used to log on to the Kafka cluster. If no authentication is enabled for the Kafka cluster, you do not need to enter the username.
Destination	Dat abase Password	The password of the username. If no authentication is enabled for the Kafka cluster, you do not need to enter the password.
Instance Details	Kafka Version	The version of the destination Kafka cluster.
	Encryption	Select <b>Non-encrypted</b> or <b>SCRAM-SHA-256</b> based on your business and security requirements.
	Торіс	Click <b>Get Topic List</b> and select a topic name from the drop-down list.
	Topic That Stores DDL Information	Select a topic from the drop-down list. The topic is used to store the DDL information. If you do not specify this parameter, the DDL information is stored in the topic that is specified by the Topic parameter.

Section	Parameter	Description
	Use Kafka Schema Registry	<ul> <li>Kafka Schema Registry provides a serving layer for your metadata. It provides a REST ful API operation to store and retrieve your Avro schemas.</li> <li>No: does not use Kafka Schema Registry.</li> <li>Yes: uses Kafka Schema Registry. In this case, you must enter the URL or IP address that is registered in Kafka Schema Registry for your Avro schemas.</li> </ul>

#### 7.

#### 8. Select the objects to synchronize.

1.Select Source and Destination	$\rightarrow$	>	4.Precheck
Synchronization Mode:One-Way Synchronization Data format delivered to Kafka:  DTS Avro Canal Json Synchronize to Kafka Partition Policy: Post all to Partition 0 Delivered different parts by hash value of lib Delivered different parts by hash value of pri Note: After the data synchronization operation i synchronization will fail	rary name + table name mary key s officially started, do not modify the	number of partitions of the target to	pic, otherwise the
Available Expand the tree before you perform a glo Q C C C C C C C C C C C C C C C C C C C	Selected (To e Edit.) Learn m dtstest04	dit an object name or its filter, hover ore. 415 Source Database Name (10bje 115 Source Table Na	over the object and dick
Select All	Select All		
<ul> <li>Rename Databases and Tables:</li> <li>Do Not Change Database and Tables:</li> <li>Retry Time for Failed Connection</li> <li>720</li> <li>Minutes</li> </ul>	nd Table Names 🔿 Change Da	tabase and Table Names	
		Cance	el Previous Next

Parameter	Description
Data Format in Kafka	The data that is synchronized to the Kafka cluster is stored in the Avro or Canal JSON format. For more information, see Data formats of a Kafka cluster.

Description
The policy used to synchronize data to Kafka partitions. Select a policy based on your business requirements. For more information, see Specify the policy for synchronizing data to Kafka partitions.
Select one or more tables from the <b>Available</b> section and click the > icon to add the tables to the <b>Selected</b> section.
<b>Note</b> DTS maps the table names to the topic name that you select in Step 6. If you want to rename the topic, you can use the object name mapping feature. For more information, see <b>Rename an object to be synchronized</b> .
You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.
By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

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

#### 10. Configure initial synchronization.

1.Select Source and Destination Instances 🔪	A.Precheck					
Initial Synchronization: Imitial Schema Synchronization Initial Full Data Synchronization Note: Trigger synchronization is not supported, please Reference Document Filter options: Image Information Information phase						
	Cancel Previous Save Precheck					
Setting	Description					
Initial Synchronization	Select both <b>Initial Schema Synchronization</b> and <b>Initial Full Data</b> <b>Synchronization</b> . DTS synchronizes the schemas and historical data of the required objects and then synchronizes incremental data.					
Filter optionsIgnore DDL in incremental synchronization phase is selected by default. In this case, DTS does not synchronize DDL operations that an performed on the source database during incremental data synchronice						

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.
- 12. Close the **Precheck** dialog box after the following message is displayed: **Precheck Passed**. Then, the data synchronization task starts.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1	>

## 7.14. Synchronize data from a selfmanaged MySQL database hosted on ECS to a self-managed Kafka cluster

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

#### Prerequisites

- A Kafka cluster is created and the Kafka version is 0.10.1.0 to 2.7.0.
- An ApsaraDB RDS for MySQL instance is created. For more information, see Create an ApsaraDB RDS for MySQL instance.

#### Precautions

• DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.

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

#### Limits

- Only tables can be selected as the objects to synchronize.
- DTS does not synchronize the data in a renamed table to the destination Kafka cluster. This applies if the new table name is not included in the objects to synchronize. If you want to synchronize the data in a renamed table to the destination Kafka cluster, you must **reselect the objects to be synchronized**. For more information, see Add an object to a data synchronization task.

## Supported synchronization topologies

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

#### Before you begin

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

#### Procedure

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

**?** Note On the buy page, set Source Instance to MySQL, Destination Instance to Kafka, 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.
| Synchronization Task Name:   | MySQL_TO_Kafka   |                 |        |                        |
|------------------------------|--|-----------------|--------|------------------------|
| Source Instance Details      |  |                 |        |                        |
|                              |  |                 |        |                        |
| Instance Type:               | User-Created Database in ECS Instance                    |                 |        |                        |
| Instance Region:             | China (Hangzhou)   |                 |        |                        |
| * ECS Instance ID:           | -  |                 |        |                        |
| Database Type:               | MySQL  |                 |        |                        |
| * Port Number:               | 3306   |                 |        |                        |
| * Database Account:          | dtstest  |                 |        |                        |
| * Database Password:         | ······ (\$>  |                 |        |                        |
|                              |  |                 |        |                        |
| Destination Instance Details |  |                 |        |                        |
|                              |  |                 |        |                        |
| Instance Type:               | User-Created Database in ECS Instance 🗸                  |                 |        |                        |
| Instance Region:             | China (Hangzhou)   |                 |        |                        |
| * ECS Instance ID:           |  |                 |        |                        |
| Database Type:               | Kafka 🗸 🗸  |                 |        |                        |
| * Port Number:               | 9092   |                 |        |                        |
| Database Account:            |  | Optional        |        |                        |
| Database Password:           | ٩)   | Optional        |        |                        |
| * Topic:                     | dtstesttopic 🗸   | Get Toplic list |        |                        |
|                              | Click Get Topic List and then select the specific topic. |                 |        |                        |
| * Kafka Version              | 0.10 ~   |                 |        |                        |
| * Encryption:                | Non-encrypted O SCRAM-SHA-256                            |                 |        |                        |
|                              |  |                 |        |                        |
|                              |  |                 | Grand  | Cet Whitelist and Next |
|                              |  |                 | Cancel | Set whitelist and Next |

Section	Parameter	Description
None	Synchronizat ion Task Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.
	lnstance Type	Select User-Created Database in ECS Instance.
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.
	ECS Instance ID	The ID of the ECS instance that hosts the source MySQL database.
	Database Type	This parameter is set to <b>MySQL</b> and cannot be changed.
Source Instance Details	Port Number	The service port number of the self-managed MySQL database. Default Value: 3306.
	Database Account	The account of the self-managed MySQL database. The account must have the SELECT permission on the required objects and the REPLICATION CLIENT, REPLICATION SLAVE, and SHOW VIEW permissions.

Section	Parameter	Description		
	Database Password	The password of the database account.		
	lnst ance T ype	The instance type of the Kafka cluster. In this example, <b>User-</b> <b>Created Database in ECS Instance</b> is selected for this parameter.		
		<b>Note</b> If you select other instance types, you must deploy the network environment for the Kafka cluster. For more information, see <b>Preparation overview</b> .		
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.		
	ECS Instance ID	The ID of the ECS instance on which the Kafka cluster is deployed.		
Destination	Dat abase T ype	Select <b>Kafka</b> .		
Instance Details	Port Number	The service port number of the Kafka cluster. Default value: 9092.		
	Dat abase Account	The username that is used to log on to the Kafka cluster. If no authentication is enabled for the Kafka cluster, you do not need to enter the username.		
	Dat abase Password	The password that corresponds to the username. If no authentication is enabled for the Kafka cluster, you do not need to enter the password.		
	Торіс	Click <b>Get Topic List</b> and select a topic name from the drop-down list.		
	Kafka Version	The version of the destination Kafka cluster.		
	Encryption	Select <b>Non-encrypted</b> or <b>SCRAM-SHA-256</b> based on your business and security requirements.		

#### 7.

8. Select the objects to synchronize.

1.Select Source and Destination		>		$\rangle$	4.Precheck
Synchronization Mode:One-W Data format delivered to Kafka : • • DT Synchronize to Kafka Partition Policy:• Pos O Del O Del Note: synchr	/ay Synchronization 5 Avro Canal Json ? t all to Partition 0 ? ivered different parts by hash value of libra ivered different parts by hash value of prin After the data synchronization operation is onization will fail	ary name + table n nary key officially started, d	ame io not modify the number of partition	is of the target topic	, otherwise the
Available Expand the tree before you per in chw in Tables in chw02 in test_polar2	form a glo	> <	Selected (To edit an object name of Edit.) Learn more.	er its filter, hover ow	er the object and dick
*Rename Databases and Tables: * Retry Time for Failed Connection	Do Not Change Database an     720 Minutes	d Table Names	<ul> <li>Change Database and Table Na</li> </ul>	ames	
				Cancel	Previous Next
Parameter	Description				
Data Format in Kafka	The data that is sync Canal JSON format. F <mark>cluster</mark> .	chronized For more ii	to the Kafka cluster nformation, see Dat	r is stored a formats	in the Avro or <mark>of a Kafka</mark>

Data Format in Kafka	The data that is synchronized to the Kafka cluster is stored in the Avro or Canal JSON format. For more information, see Data formats of a Kafka cluster.		
Policy for Shipping Data to Kafka Partitions	The policy used to synchronize data to Kafka partitions. Select a policy based on your business requirements. For more information, see Specify the policy for synchronizing data to Kafka partitions.		
	Select one or more tables from the <b>Available</b> section and click the > icon to add the tables to the <b>Selected</b> section.		
Select the objects to synchronize	<b>Note</b> DTS maps the table names to the topic name that you select in Step 6. If you want to rename the topic, you can use the object name mapping feature. For more information, see <b>Rename an object to be synchronized</b> .		
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see		
	Object name mapping.		

Parameter	Description
-----------	-------------

Retry Time for Failed	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.		
Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.		

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

#### 10. Configure initial synchronization.

1.Select Source and Destination Instances >	A.Precheck
Initial Synchronization: 🗹 Initial Filter options: 🗹 Ignore	Schema Synchronization 🛛 Initial Full Data Synchronization Note: Trigger synchronization is not supported, please Reference Document DDL in incremental synchronization phase
	Cancel Previous Save Precheck
Setting	Description
Initial Synchronization	Select both <b>Initial Schema Synchronization</b> and <b>Initial Full Data</b> <b>Synchronization</b> . DTS synchronizes the schemas and historical data of the required objects and then synchronizes incremental data.
Filter options	<b>Ignore DDL in incremental synchronization phase</b> is selected by default. In this case, DTS does not synchronize DDL operations that are performed on the source database during incremental data synchronization.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.
- 12. Close the **Precheck** dialog box after the following message is displayed: **Precheck Passed**. Then, the data synchronization task starts.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1	>

# 7.15. Synchronize data from a selfmanaged MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to a selfmanaged Kafka cluster

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

#### Prerequisites

- A Kafka cluster is created and the Kafka version is 0.10.1.0 to 2.7.0.
- The engine version of the self-managed MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The self-managed MySQL database is connected to an Alibaba Cloud virtual private cloud (VPC). For more information, see Connect an on-premises database to DTS by using CEN.

#### Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination database may contain duplicate data records.

# Limits

- Only tables can be selected as the objects to synchronize.
- DTS does not synchronize the data in a renamed table to the destination Kafka cluster. This applies if the new table name is not included in the objects to synchronize. If you want to synchronize the data in a renamed table to the destination Kafka cluster, you must **reselect the objects to be synchronized**. For more information, see Add an object to a data synchronization task.

# Supported synchronization topologies

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

# Before you begin

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

# Procedure

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

**Note** On the buy page, set Source Instance to MySQL, Destination Instance to Kafka, 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.

Source Instance Details				
Instance Type:	User-Created Database Connected Over Express Connect, VPt $\checkmark$	Guide		
Instance Region:	China (Hangzhou)			
* Peer VPC:	vpc-	Proprietary network of Other Apsara Stack A	counts	
Database Type:	MySQL			
* IP Address:	172.16			
* Port Number:	3306			
* Database Account:	dtstest			
* Database Password:	······ Ø			
Destination Instance Details				
Destination instance Details				
Instance Type:	User Created Detailers in ECC Instance			
instance type.	User-created Database in ECS Instance			
Instance Region:	China (Shanghai)			
* ECS Instance ID:	-			
Database Type:	Kafka 🗸			
* Port Number:	9092			
Database Account:		Optional		
Database Password:	٩)	Optional		
* Topic:	dtstesttopic 🗸	Get Toplic list		
	Click Get Topic List and then select the specific topic.			
* Kafka Version	1.0 ~			
* Encryption:	Non-encrypted O SCRAM-SHA-256			
			Cancel	Set Whitelist and Next

Section	Parameter	Description
None	Synchronizat ion Task Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.
	lnstance Type	Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway.
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.
	Peer VPC	The ID of the VPC that is connected to the self-managed MySQL database.
	Database Type	This parameter is set to <b>MySQL</b> and cannot be changed.
	IP Address	The server IP address of the self-managed MySQL database.
Source Instance Details	Port Number	The service port number of the self-managed MySQL database. Default value: 3306.

Section	Parameter	Description
	Database Account	The account of the self-managed MySQL database. The account must have the SELECT permission on the required objects and the REPLICATION CLIENT, REPLICATION SLAVE, and SHOW VIEW permissions.
	Dat abase Password	The password of the database account.
		The instance type of the Kafka cluster. In this example, <b>User-</b> <b>Created Database in ECS Instance</b> is selected for this parameter.
	Instance Type	<b>Note</b> If you select other instance types, you must deploy the network environment for the Kafka cluster. For more information, see <b>Preparation overview</b> .
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.
	ECS Instance ID	The ID of the Elastic Compute Service (ECS) instance on which the Kafka cluster is deployed.
Destination	Database Type	Select Kafka.
Instance Details	Port Number	The service port number of the Kafka cluster. Default value: 9092.
	Dat abase Account	The username that is used to log on to the Kafka cluster. If no authentication is enabled for the Kafka cluster, you do not need to enter the username.
	Dat abase Password	The password of the username. If no authentication is enabled for the Kafka cluster, you do not need to enter the password.
	Торіс	Click <b>Get Topic List</b> and select a topic name from the drop-down list.
	Kafka Version	The version of the destination Kafka cluster.
	Encryption	Select <b>Non-encrypted</b> or <b>SCRAM-SHA-256</b> based on your business and security requirements.

# 7.

8. Select the objects to synchronize.

1.Select Source and Destination	>		A.Preche	eck
Synchronization Mode:One-V Data format delivered to Kafka: @ DT Synchronize to Kafka Partition Policy:@ Pos O Del Note: synchr	/ay Synchronization 5 Avro Canal Json t all to Partition 0 ivered different parts by hash value of libra ivered different parts by hash value of prin After the data synchronization operation is ronization will fail	ary name + table r nary key officially started, o	ame to not modify the number of partitions of the target topic, otherwise th	e
Available Expand the tree before you pe in chw in Tables in product test_polar2	form a glo I Q	> <	Selected (To edit an object name or its filter, hover over the object a Edit.) Learn more.	and dick
Select All			Solart All	
*Rename Databases and Tables: * Retry Time for Failed Connection	Do Not Change Database and     720 Minutes	d Table Names	<ul> <li>Change Database and Table Names</li> </ul>	
			Cancel Previo	Next
Parameter	Description			
Data Format in	The data that is sync	chronized	to the Kafka cluster is stored in the A	vro or

Parameter	Description			
Data Format in Kafka	The data that is synchronized to the Kafka cluster is stored in the Avro or Canal JSON format. For more information, see Data formats of a Kafka cluster.			
Policy for Shipping Data to Kafka Partitions	The policy used to synchronize data to Kafka partitions. Select a policy based on your business requirements. For more information, see Specify the policy for synchronizing data to Kafka partitions.			
	Select one or more tables from the <b>Available</b> section and click the > icon to add the tables to the <b>Selected</b> section.			
Select the objects to synchronize	<b>Note</b> DTS maps the table names to the topic name that you select in Step 6. If you want to rename the topic, you can use the object name mapping feature. For more information, see <b>Rename an object to be synchronized</b> .			

Description			
You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.			
By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.			
<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.			

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

#### 10. Configure initial synchronization.

1.Select Source and Destination Instances >	4.Precheck
Initial Synchronization: 🗹 Initial Filter options: 🗹 Ignore	Schema Synchronization 🗹 Initial Full Data Synchronization Note: Trigger synchronization is not supported, please Reference Document DDL in incremental synchronization phase
	Cancel Previous Save Precheck
Setting	Description
Initial Synchronization	Select both <b>Initial Schema Synchronization</b> and <b>Initial Full Data</b> <b>Synchronization</b> . DTS synchronizes the schemas and historical data of the required objects and then synchronizes incremental data.
Filter options	<b>Ignore DDL in incremental synchronization phase</b> is selected by default. In this case, DTS does not synchronize DDL operations that are performed on the source database during incremental data synchronization.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.
- 12. Close the **Precheck** dialog box after the following message is displayed: **Precheck Passed**. Then, the data synchronization task starts.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1 >	*

# 7.16. Synchronize data from an ApsaraDB RDS for MySQL instance to a PolarDB-X instance

PolarDB-X is developed by Alibaba Cloud to address the bottleneck of single-host database services. DRDS is compatible with the MySQL protocol and syntax, and supports automatic sharding, online smooth scaling, auto scaling, and transparent read and write splitting. DRDS provides O&M capabilities throughout the lifecycle of databases. This topic describes how to synchronize data from an ApsaraDB RDS for MySQL instance to a PolarDB-X instance by using Data Transmission Service (DTS).

#### Prerequisites

- The tables to be synchronized from the source database contain primary keys.
- The destination database has sufficient storage space.
- A PolarDB-X instance is created. For more information, see Create a instance and Create a database.

**?** Note When you create an instance, you must select RDS MySQL as the storage type.

#### Precautions

• DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.

- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source database. The destination database has sufficient storage space.
- DTS does not synchronize schemas from an ApsaraDB RDS for MySQL instance to a PolarDB-X instance. Before you configure a data synchronization task, you must create databases and tables in the destination instance.

### Limits

- You can select only tables as the objects to be synchronized.
- DTS does not synchronize the following types of data: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID\_SNAPSHOT.
- Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to 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.

# SQL operations that can be synchronized

INSERT, UPDATE, and DELETE

# Permissions required for database accounts

Database	Required permissions
ApsaraDB RDS for MySQL	The SELECT permission on the objects to be synchronized, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission
PolarDB-X	DTS automatically creates a database account and grants permissions to the account. You do not need to specify the database account.

# Supported synchronization topologies

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

# Before you begin

When you synchronize data from an ApsaraDB RDS for MySQL instance to a PolarDB-X instance, note that DTS does not support **initial schema synchronization**. Therefore, you must create databases and tables in the destination instance based on the schemas of the objects in the source ApsaraDB RDS for MySQL instance. For more information, see Create a database and Create a table.

**Note** During schema synchronization, DTS synchronizes the schemas of the required objects from the source database to the destination database.

# Procedure

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

**Note** On the buy page, set Source Instance to MySQL, set Destination Instance to DRDS, and set Synchronization Topology to One-way Synchronization.

- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click **Data Synchronization**.
- 4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.
- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

1.Select Source and Destination Insta	ances for 2.Select Object to Be Synchronized	$\geq$	3.Advanced Settings	$\rangle$	4.1	Precheck
Synchronization Task Name:	rdsmysql_to_drds					
Source Instance Details						
Instance Type:	RDS Instance	~				
Instance Region:	China (Hangzhou)					
* Instance ID:	rm-	•	RDS Instances of Other Apsara Stack Accounts			
* Database Account:	dtstest					
* Database Password:	••••••	¢>				
* Encryption:	Non-encrypted OSSL-encrypted					
Destination Instance Details						
Instance Type:	DRDS Instance					
Instance Region:	China (Hangzhou)					
DRDS Instance ID:	drde	•				
					Cancel	Set Whitelist and Nevt

Section	Parameter	Description
N/A	Synchronizatio n Task Name	DTS automatically generates a task name. We recommend that you specify an informative name to identify the task. You do not need to use a unique task name.
	Instance Type	Select RDS Instance.
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.
	Instance ID	Select the ID of the source ApsaraDB RDS for MySQL instance.

Section	Parameter	Description		
Source Instance Details		Enter the database account of the source RDS instance. For information about the permissions that are required for the account, see Permissions required for database accounts.		
	Database Account	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 of the database account.			
		Select <b>Non-encrypted</b> or <b>SSL-encrypted</b> . If you want to select <b>SSL-encrypted</b> , 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 ApsaraDB RDS for MySQL instance.		
		<b>Notice</b> The <b>Encryption</b> parameter is available only for regions in the Chinese mainland and the China (Hong Kong) region.		
	Instance Type	This parameter is set to <b>DRDS Instance</b> and cannot be changed.		
Destination Instance Details	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.		
Details	DRDS Instance ID	Select the ID of the destination DRDS instance.		

7.

8. Select the synchronization policy and the objects to be synchronized.

1.Select Source and Destination	2.Select Object to Be Synchronized     3.Advanced Settings     4.Precheck
Synchronization Mode:	One-Way Synchronization
Available Expand the tree before y Control chw Control	Selected (To edit an object name or its filter, hover over the object and dick Edit;) Learn more.     Hover over the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object name or its filter, hover over the object and dick the dalog biopears, may object and dick the dalog biopears, filter, hover over the object and dick the dalog biopears, may object and dick the dalog biopears, filter, hover over the object and dick the dalog biopears, filter, hover over the object and dick the dalog biopears, may object and dick the dalog biopears, filter, hover over the object and dick the dalog biopears, filter, hover over the object and dick the dalog biopears, may object and dick the dalog biopears, may object and dick the dalog biopears
Select All	Select All
*Rename Databases and Tal	Image: Dot Not Change Database and Table Names       O       Change Database and Table Names         Image: Dot Not Change Database and Table Names       Image: Dot Not Change Database and Table Names         Image: Dot Not Change Database and Table Names       Image: Dot Not Change Database and Table Names         Image: Dot Not Change Database and Table Names       Image: Dot Not Change Database and Table Names         Image: Dot Not Change Database and Table Names       Image: Dot Not Change Database and Table Names         Image: Dot Not Change Database and Table Names       Image: Dot Not Change Database and Table Names         Image: Dot Not Change Database and Table Names       Image: Dot Not Change Database and Table Names         Image: Dot Not Change Database and Table Names       Image: Dot Not Change Database and Table Names         Image: Dot Not Change Database and Table Names       Image: Dot Not Change Database and Table Names         Image: Dot Not Change Database and Table Names       Image: Dot Not Change Database and Table Names         Image: Dot Not Change Database and Table Names       Image: Dot Not Change Database and Table Names         Image: Dot Not Change Database And Table Names       Image: Dot Not Change Database And Table Names         Image: Dot Not Change Database And Table Names       Image: Dot Names         Image: Dot Not Change Database And Table Names       Image: Dot Names         Image: Dot Not Change Database And Table Names       Image: Do
Setting	Description
Select the objects to be synchronized	<ul> <li>Select one or more tables from the Available section and click the &gt; icon to move the tables to the Selected section.</li> <li>Note <ul> <li>You can select only tables as the objects to be synchronized.</li> <li>By default, after an object is synchronized to the destination instance, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination, see Rename an object to be synchronized.</li> </ul></li></ul>
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.

Setting	Description
	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
Retry Time for Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

#### 9. Click Next.

10. Specify whether you want to perform initial full data synchronization.

		3.Advanced Settings	4.Precheck
Initial Synchronization: Initial Full Dat	ta Synchronization		
			Cancel Previous Save Precheck

**?** Note During initial full data synchronization, DTS synchronizes the historical data of the required objects from the source database to the destination database. If you do not select Initial Full Data Synchronization, DTS does not synchronize the historical data.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscription Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < <b>1</b> > »

# 8.Synchronize data to or from an Apsara PolarDB for MySQL cluster

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

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

# Prerequisites

- The source and destination PolarDB for MySQL clusters are created. For more information, see Create a PolarDB for MySQL cluster.
- The binary logging feature is enabled for the source and destination PolarDB for MySQL clusters. 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 loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After the initial full data synchronization is completed, the tablespace of the destination cluster is larger than that of the source cluster.
- The source cluster must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, the destination cluster may contain duplicate data records.
- If the source or destination instance is located in a region outside the Chinese mainland, two-way data synchronization is supported only between instances located within the same region. For example, two-way data synchronization is supported between instances within the Japan (Tokyo) region. Two-way data synchronization between an instance in the Japan (Tokyo) region and another instance in the Germany (Frankfurt) region is not supported.

# Limits

• DTS supports two-way data synchronization between two PolarDB clusters, but not between multiple PolarDB clusters.

- •
- Incompatibility with triggers
- If you select a database as the object to synchronize and the database contains a trigger that updates a table, data inconsistency may occur. For more information about how to solve this issue, see Configure a data synchronization task for a source database that contains a trigger.
- Limits on RENAME TABLE operations RENAME TABLE operations may cause data inconsistency between the source and destination databases. For example, if you select a table as the object and rename the table during data synchronization, the data of this table is not synchronized to the destination database. To prevent 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 two-way data synchronization, you can synchronize DDL operations only in one direction. If DDL synchronization in a direction is configured, DDL synchronization in the opposite direction is not supported. Only DML operations can be synchronized in the opposite direction.

# SQL operations that can be synchronized

# **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 a single PolarDB cluster. If data records are updated on both two PolarDB clusters, a synchronization conflict occurs.

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

• Uniqueness conflicts caused by INSERT operations

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

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

• Non-existent records to be deleted The records to delete do not exist in the destination cluster. In this case, DTS ignores the DELETE

operation regardless of the conflict resolution policy that you specify.

#### ♥ 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 ensure that the conflict detection mechanism can prevent all data conflicts. To perform two-way synchronization, make sure that records with the same primary key, business primary key, or unique key are updated only on one of the synchronization nodes.
- DTS provides conflict resolution policies to prevent conflicts that may occur during data synchronization. You can select a conflict resolution policy when you configure two-way data synchronization.

### Procedure

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

Notice On the buy page, set both Source Instance and Destination Instance to PolarDB and set Synchronization Topology to Two-way Synchronization.

- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click **Data Synchronization**.
- 4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.
- 5. Configure the data synchronization task in the forward direction.
  - i. Find the data synchronization instance, and click **Configure Task** in the **Actions** column of the first data synchronization task.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
			Pay-As-You-Go	Two-Way Synchronization	Switch to Subscription   Upgrade View Synchronization Task ~   More
Task Name	Status	Synchronization Details	Source/Destination Instance		Actions
	Not Configured		Not Configured Not Configured	Co	nfigure Synchronization Channel
	Not Configured		Not Configured Not Configured	Co	nfigure Synchronization Channel

ii. Configure the source and destination clusters.

1.Configure Source and Destinatio	n 2.Select Objects to Synchronize	$\geq$	3.Advanced Settings	> 4	I.Precheck
Synchronization Task Name:	PolarDB MySOL Forward				
_					
Source Instance Details					
Instance Type:	PolarDB Instance	•			
Instance Region:	China (Hangzhou)				
* PolarDB Instance ID:	pc-bp.	•			
* Database Account:	dtetast				
* Database Password:	*******	<⊅			
Destination Instance Details					
Instance Type:	PolarDB Instance	~			
Instance Region:	China (Hangzhou)				
PolarDB Instance ID:	pc-bp	-			
I Database Associate					
" Database Account:	dtstest				
* Database Password:	••••••	<⊅			
				Cancel	Set Whitelist and Next

Section	Parameter	Description
None	Synchroniza tion Task Name	The task name that DTS generates. We recommend that you specify a descriptive name that makes it easy to identify. You do not need to use a unique task name.
	Instance Type	The instance type of the source instance. The value of this parameter is set to <b>PolarDB Instance</b> and cannot be changed.
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.
	PolarDB Instance ID	The ID of the source PolarDB for MySQL cluster.
Source Instance Details		<b>Notice</b> When you configure the data synchronization task in the reverse direction, select the ID of the destination PolarDB cluster of the forward synchronization.
	Database Account	The database account of the source PolarDB cluster. The account must have the read and write permissions on the objects to synchronize.
	Database Password	The password of the database account.
	lnstance Type	The instance type of the destination instance. The value of this parameter is set to <b>PolarDB Instance</b> and cannot be changed.
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.

Section	Parameter	Description		
<b>2</b>	PolarDB Instance ID	The ID of the destination PolarDB for MySQL cluster.		
Destination Instance Details		<b>Notice</b> When you configure the data synchronization task in the reverse direction, select the ID of the source PolarDB cluster of the forward synchronization.		
	Dat abase Account	The database account of the destination PolarDB cluster. The account must have the read and write permissions on the objects to synchronize.		
	Database Password	The password of the database account.		

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

#### ? Note

- You do not need to modify the security settings for ApsaraDB instances (such as and ) and ECS-hosted databases. DTS automatically adds the CIDR blocks of DTS servers to the whitelists of ApsaraDB instances or the security group rules of Elastic Compute Service (ECS) instances. For more information, see Add the CIDR blocks of DTS servers to the security settings of on-premises databases.
- After data synchronization is completed, we recommend that you remove the CIDR blocks of DTS servers from the whitelists or security groups.
- iv. Select the synchronization policy and the objects to synchronize.

1.Select Source and	Destination > 2.Select 0	bject to Be Synchronized	3.Advanced Settings	>	4.Precheck
Synchronization	Mode:Two-Way Synchronization	(DML+DDL)			
Fundado D.D.L. Obstance					
Exclude DDL Stateme	nts:) Yes 💿 No				
DML Stateme Synchroni	ents for ization: 🖌 Insert 🛛 🖌 Delet	te 🔽 Update			
Conflict Resolution Po	olicy: TaskFailed (When a confli	ct occurs, an error is reported and the	task 🗸		
Available			Selected (To edit an object n Edit.) Learn more.	ame or its filter, ho	over over the object and click
Expand the tree b	efore you perform a glo				
🗄 🦮recycle_t	bin				Q
🕀 📂 asd			📔 dtstestdata		
🕂 📂 chw02					
⊕ 🚰 dts		>			
tstest0512	2_Jznz_0001_ext_0001	/			
dtstest125	1	<			
+ 👝 sys	-				
Colort All					
Select All			Select All		
*Rename Databases	and Tables: 💿 Do No	t Change Database and Table Names	O Change Database and Ta	ble Names	
*Source table DMS_ ( want to copy the tem the target database d	ONLINE_ Do you O Yes porary table to luring DDL:	No ?			
* Retry Time for Faile	ed Connection 720	Minutes 🕐			
				Cancel	Previous
tting	Parameter	Description			
		To exclude DDL	operations, selec	t Yes.	

	<ul> <li>To exclude DDL operations, select Yes.</li> <li>To include DDL operations, select No.</li> </ul>
Exclude DDL Statements	<b>Notice</b> Limits on DDL synchronization direction: To ensure the stability of two-way data synchronization, you can synchronize DDL operations only in the forward direction.
DML Statements for Synchronizati on	Select the types of DML operations that you want to synchronize. By default, the <b>INSERT</b> , <b>UPDATE</b> , and <b>DELETE</b> operations are selected. You can select the DML operation types based on your business requirements.

Setting	Parameter	Description
	Conflict Resolution Policy	<ul> <li>Select the resolution policy for synchronization conflicts. By default, <b>TaskFailed</b> is selected. You can select a conflict resolution policy based on your business requirements.</li> <li><b>TaskFailed</b> 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. </li> <li><b>Ignore</b> 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. </li> <li><b>Overwrite</b> If a conflict occurs during data synchronization, the conflicting records in the destination database are used. </li> </ul>
Select the synchronizatio n policy		

Setting	Parameter	Description Select one or more objects (tables or a database) from the				
Select the objects to be synchronized	N/A	<ul> <li>Available section and click the &gt; icon to move the objects to the Selected section.</li> <li>Notice <ul> <li>If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination instance, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Rename an object to be synchronized.</li> </ul> </li> </ul>				
Rename Databases and Tables	N/A	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.				
Replicate Temporary Tables When DMS Performs DDL Operations	N/A	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> <li>Yote If online DDL operations generate a large amount of data, the data synchronization task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> <li>Note If you select No, the tables in the destination database may be locked.</li> </ul>				

Setting	Parameter	Description
Retry Time for		By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
Failed Connections	N/A	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

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

1.Configure Source and Destination	> 2	Select Objects	s to Synchronize		3.Advanced Settings		4.Precheck	
Initial Synchronization: 🗹	Initial Schema Syn preverse synchror	nchronization nization will not ge	✓ Initial Full Data Syn o through the same proc	chronization ess during forw	Note: Trigger synchronization is ard synchronization.)	not supported, please	e Reference Document	(Tables that
						Cancel	Previous Save	Precheck
<u>_</u>								

#### ♥ Notice

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

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

? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed

item to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.
- viii. Close the **Precheck** dialog box after the following message is displayed: **Precheck Passed**. Then, the data synchronization task in the forward direction starts.
- 6. Wait until initial synchronization is completed and the data synchronization task is in the **Synchronizing** state.

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

- 7. Configure the data synchronization task in the reverse direction.
  - i. Find the second data synchronization task, and click **Configure Task** in the Actions column.

			Pay-As-You-Go	Switch to Subscription Two-Way Upgrade Synchronization View Synchronization Task ~   More
Task Name	Status	Synchronization Details	Source/Destination Instance	Actions
PolarDB MySQL_Forward	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	pc-bp t5b pc-bp 3hf	Pause Task   More
hangzhou-hangzhou-medium	Not Configured		pc-bp hf pc-bp 5b	Configure Synchronization Channel

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

#### Result

After a period of time, both data synchronization tasks are in the **Synchronizing** state.

Syn	chronization Task ID 🔻		Search Sort: Default So	rting 🔻 Status: All	v	
	Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
				Pay-As-You-Go	Two-Way Synchronization	Switch to Subscription Upgrade View Synchronization Task × More
	Task Name	Status	Synchronization Details	Source/Destination Instance		Actions
	PolarDB MySQL_Forward	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	pc-bp t5b pc-bp 3hf		Pause Task   More
	PolarDB MySQL_Reverse 📝	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	pc-bp 3hf pc-bp 1t5b		Pause Task   More

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

PolarDB is a next-generation relational database service that is developed by Alibaba Cloud. PolarDB is compatible with the MySQL database engine and features high availability, ease of use, and reliability. This topic describes how to configure one-way data synchronization between PolarDB for MySQL clusters by using Data Transmission Service (DTS).

# Prerequisites

- The source and destination PolarDB for MySQL clusters are created. For more information, see Purchase a pay-as-you-go cluster. For more information about the supported database engine versions, see Overview of data synchronization scenarios.
- The binary logging feature is enabled for the source 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 loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization is complete, the size of used 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, the destination cluster may contain duplicate data records.
- To ensure compatibility, the database engine version of the destination PolarDB for MySQL cluster must be the same as or later than that of the source PolarDB for MySQL cluster.

# SQL operations that can be synchronized

# 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, see Synchronization topologies.

# Limits

• Incompatibility with triggers

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

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

# Procedure

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

**Note** On the buy page, set Source Instance and Destination 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.

1.Configure Source and	Destination	n 💙 2.5	elect Objects to Synchronize	$\rightarrow$	3.Advanced Settings	>	4.Precheck
Synchronization	Task Name:	PolarDB_TO_PolarDB	3				
Source Instance Details							
Inst	ance Type:	PolarDB Instance		•			
Instance Region: Singapore							
* PolarDB II	nstance ID:	pc-gs5		•			
* Databa	se Account:	dtstest					
* Database	Password:	•••••		<b>♦</b> >			
Destination Instance Details							
Inst	tance Type:	PolarDB					
Instar	nce Region:	Singapore					
* PolarDB I	nstance ID:	pc-gs5					
* Databa	se Account:	dtstest					
* Database	Password:						
							_
							Cancel Set Whitelist and Next
Section	Dara	motor	Description				
Section	Pdid	meter	Description				

Section	Parameter	Description
None	Synchronizat ion Task Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.
	lnstance Type	The value of this parameter is set to <b>PolarDB Instance</b> and cannot be changed.
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.
Source	PolarDB Instance ID	The ID of the source PolarDB for MySQL cluster.
Instance		The database account of the source PolarDB for MySQL cluster.
Details	Database Account	<b>Note</b> The account must have the SELECT permission on the objects to synchronize and the REPLICATION CLIENT, REPLICATION SLAVE, and SHOW VIEW permissions.
	Dat abase Password	The password of the database account.
	lnstance Type	This parameter is set to <b>PolarDB</b> and cannot be changed.
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.
Destination	PolarDB Instance ID	The ID of the destination PolarDB for MySQL cluster.
Instance Details		The database account of the destination PolarDB for MySQL cluster.
	Database Account	<b>Note</b> The database account must have the ALL permission on the objects to synchronize.
	Dat abase Password	The password of the database account.

7.

8. Select the processing mode of conflicting tables and the objects to be synchronized.

		Selected (To edit an object nam	ne or its filter, hover ov	er the object and click
Expand the tree before you perfo	rm a glol 🛛 🔍	Edit.) Learn more.		
			Q	
🕂 🧁 asd		📴 dtstestdata		
e chw02  f chw02  f chw02  f chw02				
dtstest0512_jzhz_0001_ext	£_0001			
🕀 🦢 dtstest123				
🕀 🦢 dtstestdata1	<			
-/-				
Select All				
Select All		Select All		
Select All *Rename Databases and Tables:	<ul> <li>Do Not Change Database and Table Nan</li> </ul>	Select All es O Change Database and Table	e Names	
Select All *Rename Databases and Tables: *Source table DMS_ONLINE_ Do you want to copy the temporary table to prove the temporary table to be te	<ul> <li>Do Not Change Database and Table Nan</li> <li>Yes</li> <li>No</li> </ul>	Select All es	e Names	
Select All  Rename Databases and Tables:  Source table DMS_ONLINE_ Do you want to copy the temporary table to the target database during DDL:	<ul> <li>Do Not Change Database and Table Nan</li> <li>Yes</li> <li>No</li> </ul>	Select All Select All Change Database and Table	e Names	
Select All  *Rename Databases and Tables: *Source table DMS_ONLINE_ Do you want to copy the temporary table to the target database during DDL: * Retry Time for Failed Connection	Do Not Change Database and Table Nan u Yes No 720 Minutes 7	Select All Select All Change Database and Table	e Names	
Select All  Rename Databases and Tables:  Source table DMS_ONLINE_ Do yoo want to copy the temporary table to the target database during DDL:  Retry Time for Failed Connection	Do Not Change Database and Table Nan u Yes No 7 Minutes 7	Select All es Ochange Database and Table	e Names	

Setting	Description
	• <b>Pre-check and Intercept</b> : 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.
	<b>Note</b> You can use the object name mapping feature to rename the tables that are synchronized to the destination database. You can use this feature if the source and destination databases contain identical table names and the tables in the destination database cannot be deleted or renamed. For more information, see Rename an object to be synchronized.
	<ul> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> </ul>
Select the processing mode of conflicting tables	<ul> <li>Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks.</li> <li>During initial data synchronization, DTS does not synchronize the data records that have the same primary keys as the data records in the destination database. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization.</li> <li>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.</li> </ul>

Setting	Description
	Select one or more objects from the <b>Available</b> section and click the $\rightarrow$ icon to move the objects to the <b>Selected</b> section. You can select tables or databases as the objects to be synchronized.
Select the objects to be synchronized	<ul> <li>Note</li> <li>If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination cluster, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination cluster. For more information, see Rename an object to be synchronized.</li> </ul>
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.
	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> </ul>
Replicate Temporary Tables When DMS Performs DDI	<b>Note</b> If online DDL operations generate a large amount of data, the data synchronization task may be delayed.
Operations	• <b>No</b> : DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.
	Note If you select No, the tables in the destination database may be locked.
Retry Time for Failed	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

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

1.Configure Source and Destination	2.Select Objects to Sy	vnchronize	3.Advanced Settings		4.Precheck
Initial Synchronization: 🗹 Initial S	Schema Synchronization	Initial Full Data Synchronizati	on		
			Ca	ncel Previous	Save Precheck

**?** Note Initial synchronization includes initial schema synchronization and initial full data synchronization. If you select both Initial Schema Synchronization and Initial Full Data Synchronization, DTS synchronizes the schemas and historical data of the required objects before DTS synchronizes incremental data.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Action	ns
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch t Subscription Upgrad Mor	to le re
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < <b>1</b> > »	

# 8.3. Synchronize data from a PolarDB for MySQL cluster to an ApsaraDB RDS for MySQL instance

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

#### Prerequisites

- A PolarDB for MySQL cluster is created. For more information, see Create a PolarDB for MySQL cluster.
- The binary logging feature is enabled for the 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 loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.

•

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

# SQL operations that can be synchronized

# 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, see Synchronization topologies.

#### Limits

• Incompatibility with triggers

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

Limits on RENAME TABLE operations
 RENAME TABLE operations may cause data inconsistency between the source and destination
 databases. For example, if only Table A is selected as the object to synchronize and is renamed Table
 B, Table B cannot be synchronized to the destination database. To prevent this situation, you can
 select the entire database where Table A is located as the object to synchronize when you configure
 the data synchronization task.

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

**?** Note On the buy page, set Source Instance to PolarDB, Destination 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.

1.Configure Source and Destinatio	n 2.Select Objects to Synchronize	$\rightarrow$	3.Advanced Settings	A.Precheck
Synchronization Task Name:	PolarDB_TO_MySQL			
Source Instance Details				
Instance Types		_		
Instance Type.		*		
Instance Region:	Singapore			
PolarDB Instance ID:	pc-gs.	•		
* Database Account:	dtstest			
* Database Password:	*****	<b>∢</b> >		
Destination Instance Details				
Instance Type:	RDS Instance	•		
Instance Region:	Singapore			
* Instance ID:	rm-g:	•		
* Database Account:	dtstest			
* Database Password:	******	<b>∢</b> >		

Set Whitelist

Cancel

Section	Parameter	Description
None	Synchronizatio n Task Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.
	Instance Type	The instance type of the destination instance. The value of this parameter is set to <b>PolarDB Instance</b> and cannot be changed.
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.
Courco	PolarDB Instance ID	The ID of the source PolarDB for MySQL cluster.
Instance Details		

Section	Parameter	Description		
	Dat abase Account	The database account of the source PolarDB cluster.		
	Dat abase Password	The password of the database account.		
	Instance Type	Select RDS Instance.		
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.		
	Dat abase Account	The database account of the destination ApsaraDB RDS for MySQL instance.		
		The password of the database account.		
Destination Instance Details	Dat abase Password	Note If the database engine of the destination ApsaraDB RDS for MySQL instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password.		
	Encryption	Select <b>Non-encrypted</b> or <b>SSL-encrypted</b> . If you select <b>SSL-encrypted</b> , you must enable SSL encryption for the ApsaraDB RDS for MySQL instance before you configure the data synchronization task. For more information, see <b>Configure SSL encryption for an ApsaraDB RDS for MySQL instance</b> .		
		<b>Notice</b> The <b>Encryption</b> parameter is available only for regions in the Chinese mainland and the China (Hong Kong) region.		

7.

# 8. Select the processing mode of conflicting tables and the objects to be synchronized.

1.Select Source and Destination	2.Select Object to Be Synchronized	3.Advanced Settings	>	4.Precheck
Synchronization Mode: <b>One-Wa</b>	/ Synchronization (DML+DDL)			
Available		Selected (To edit an object nar Edit.) Learn more.	ne or its filter, hover o	ver the object and click
Expand the tree before you perfo	rm a glol 🛛 🔍		Q	
+ 🖕 asd + 🚰 chw02		📔 dtstestdata		
	_0001			
	<			
+ 📂 sys				
Select All				
*Rename Databases and Tables:	Do Not Change Database and Table Names	Change Database and Table	e Names	
*Source table DMS_ ONLINE_ Do yo want to copy the temporary table to the target database during DDL:	u 🔿 Yes 💿 No 🥐			
* Retry Time for Failed Connection	720 Minutes 🥐			
			Cancel	Previous
	Description			

Setting	Description
Select the processing mode of conflicting tables	<ul> <li>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.</li> <li>Note You can use the object name mapping feature to rename the tables that are synchronized to the destination database. You can use this feature if the source and destination databases contain identical table names and the tables in the destination database cannot be deleted or renamed. For more information, see Rename an object to be synchronized.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>During initial data synchronization, DTS does not synchronize the data records that have the same primary keys as the data records in the destination databases. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization.</li> <li>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 synchronized or the data synchronized or the data</li> </ul>
	Select one or more objects from the <b>Available</b> section and click the > icon to move the objects to the <b>Selected</b> section. You can select tables or databases as the objects to be synchronized.
Select the objects to be synchronized	<ul> <li>Note</li> <li>If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination cluster, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination cluster. For more information, see Rename an object to be synchronized.</li> </ul>

Setting	Description		
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.		
Replicate Temporary Tables When DMS Performs DDL Operations	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> <li>Note If online DDL operations generate a large amount of data, the data synchronize the data of temporary tables generated by online DDL operation task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> <li>Note If you select No, the tables in the destination database may be locked.</li> </ul>		
Retry Time for Failed	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.		
Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.		

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

#### 10. Select the initial synchronization types.

1.Configure Source and Destination	2	2.Select Objects to			3.Advanced Settings			4.Precheck	
Initial Synchronization: 🗹	Initial Schem	a Synchronization	☑ Initial Full Data	Synchronization					
						Cancel	Previous	Save	Precheck

Note Initial synchronization includes initial schema synchronization and initial full data synchronization. If you select both Initial Schema Synchronization and Initial Full Data Synchronization, DTS synchronizes the schemas and historical data of the required objects before DTS synchronizes incremental data.

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

? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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



# 8.4. Synchronize data from a PolarDB for MySQL cluster to a DataHub instance

DataHub is a real-time data distribution platform that is designed to process streaming data. You can publish and subscribe to streaming data in DataHub and distribute the data to other platforms. DataHub allows you to analyze streaming data and build applications based on streaming data. This topic describes how to synchronize data from a PolarDB for MySQL cluster to a DataHub instance by using Data Transmission Service (DTS). After you synchronize data to DataHub, you can use big data services such as Realtime Compute for Apache Flink to analyze data in real time.

# Prerequisites

- The DataHub instance resides in the China (Hangzhou), China (Shanghai), China (Beijing), or China (Shenzhen) region.
- A DataHub project is created to receive the synchronized data. For more information, see Create a project.
- The binary logging feature is enabled for the PolarDB for MySQL cluster. For more information, see Enable binary logging.
- The tables to be synchronized from the PolarDB for MySQL cluster have PRIMARY KEY or UNIQUE constraints.

# Limits

- Initial full data synchronization is not supported. DTS does not synchronize historical data of the required objects from the source PolarDB cluster to the destination DataHub instance.
- You can select only tables as the objects to be synchronized.
- After a data synchronization task is started, DTS does not synchronize columns that are created in the source PolarDB cluster to the destination DataHub instance.
- We recommend that you do not perform data definition language (DDL) operations on the required objects during data synchronization. Otherwise, data synchronization may fail.

# SQL operations that can be synchronized

INSERT, UPDATE, and DELETE

# Procedure

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

On the buy page, set Source Instance to PolarDB, set Destination Instance to DataHub, 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.

#### Data Synchronization Synchronize d ata to or from an Apsara PolarDB fo r MySQL cluster

1.Specify Source an	nd Destination Dat	abases >	2.Select O	bject to Be Synch	hronized	>	3.P	recheck
Synchror	nization Task Name:	hangzhou-h	angzhou-medium					
Source Instance Details	5							
	Instance Type:	PolarDB Ins	tance			~		
	Instance Region	China (Hanga	how					
	Instance Region.		nou)					
* Pc	blarDB Instance ID:	pc-	distant.			-		
						_		
	Database Account:	dtstest						
* 0	atabase Password:				<	1>		
					s			
Destination Instance D	etails							
	Instance Type:	DataHub						
	Testere Desires	China (Unanal	h)					
	Instance Region:	China (Hangzi	nou)					
	* Project:	dtstestdata				•		
	100 B.							
							Cancol	Cot Whitelist and Next
							Cancer	Sec whitelist and Next
Castian	Daramata	~	Deceription					

Section	Parameter	Description
N/A	Synchronizatio n Task Name	DTS automatically generates a task name. We recommend that you specify an informative name to identify the task. You do not need to use a unique task name.
	Instance Type	The value of this parameter is set to <b>PolarDB Instance</b> and cannot be changed.
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.
Source Instance Details	PolarDB Instance ID	Select the ID of the source PolarDB cluster.
	Dat abase Account	Enter the database account of the source PolarDB cluster.
	Dat abase Password	Enter the password of the database account.
Destination	Instance Type	The value of this parameter is set to <b>DataHub</b> and cannot be changed.
Instance Details	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.
	Project	Select the name of the DataHub <b>project</b> .

#### 7.

8. Select the synchronization policy and the objects to be synchronized.

1.Specify Source and Destina	tion Databases > 2.Select Object to Be Synchronized > 3.Precheck
Initial Synchronization: 🔽	nitial Schema Synchronization
Available	Selected (To edit an object name or its filter, hover over the object and clic Edit.) Learn more.
Expand the tree before you per asd testexcel testexcel test123 test12345	form a glol Q dstestdata (10bjects) i_ext_0001 0001
Select All	Select All
*Rename Databases and Tables:	Do Not Change Database and Table Names     O Change Database and Table Names
* Retry Time for Failed Connectio	720 Minutes 🧭
*Whether to enable new addition column rules	al O Yes O No
	Cancel Previous Precheck
Setting	Description
	Select Initial Schema Synchronization.
nitial Synchronization	<b>Note</b> After you select <b>Initial Schema Synchronization</b> , DTS synchronizes the schemas of the required objects (such as tables) to the destination DataHub instance.
	Select one or more objects from the <b>Available</b> section and click the <b>&gt;</b> icon to move the objects to the <b>Selected</b> section.
Select the objects o be synchronized	<ul> <li>Note</li> <li>You can select only tables as the objects to be synchronized.</li> <li>By default, after an object is synchronized to the destination instance, the name of the object remains unchanged. You can use</li> </ul>

Setting	Description
Whether to enable the new naming rules for additional columns	After DTS synchronizes data to DataHub, DTS adds additional columns to the destination topic. If the names of additional columns are the same as the names of existing columns in the destination topic, data synchronization fails. Select <b>Yes or No to specify whether you want to enable the new naming rules for additional columns</b> .
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.
Replicate Temporary Tables When DMS Performs DDL Operations	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> <li>Note If online DDL operations generate a large amount of data, the data synchronization task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> <li>Note If you select No, the tables in the destination database may be locked.</li> </ul>
Retry Time for Failed Connections	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.

9. (Optional)In the **Selected** section, move the pointer over the destination topic and click **Edit**. In the dialog box that appears, set the shard key. The shard key is used for partitioning.

Available	Selected (To edit an object name or its filter, hover over the object and Edit ) Learn more.
Expand the tree before you perform a glol   Q	
🕂 늘 asd	
🕂 🦢 testexcel	dtstestdata (10bjects)
+ 🖕 test123	i order
+ = test12345	
the hot_dtstest0512_jznz_0001	
The dts	
⊕ ghj_db	
etitest	
🕂 📴 dasdtest	
🕂 📴 orderinfo 🔹 👻	
Select All	
	Select All
Rename Databases and Tables: <ul> <li>Do Not Change Database and Table Names</li> </ul>	Change Database and Table Names
Retry Time for Failed Connection 720 Minutes 🧑	

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the 🕧 icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscription Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1 > »

# Schema of a DataHub topic

When DTS synchronizes incremental data to a DataHub topic, DTS adds additional columns to store metadata. The following figure shows the schema of a DataHub topic.

**Note** In this example, id, name, address are data fields. DTS adds the dts\_ prefix to data fields because the previous naming rules for additional columns are used.

dts_id	dts_name	dts_address	dts_record_id	dts_operation_flag	dts_instance_id	dts_db_name	dts_table_name	dts_utc_timestamp	dts_before_flag	dts_after_flag
10006	- <b>1</b> 11		1574832130000000000	U		dtstestdata	customer	1574832130	Y	N
10006	- 1810 - I		1574832130000000000	U		dtstestdata	customer	1574832130	N	Y
10009	1822	10010	1574832919000000000	D		dtstestdata	customer	1574832919	Y	N
10112	100		1574832919000000000	I		dtstestdata	customer	1574832919	N	Y

Previous additional column name	New additional column name	Data type	Description
dts_record _id	new_dts_sync_dts_rec ord_id	String	<ul> <li>The unique ID of the incremental log entry.</li> <li>Note</li> <li>By default, the ID auto-increments for each new log entry. In disaster recovery scenarios, rollback may occur, and the ID may not auto-increment. Therefore, some IDs may be duplicated.</li> <li>If an UPDATE operation is performed, DTS generates two incremental log entries to record the pre-update and post-update values. The values of the dts_record_id_field in the two incremental log entries are the same.</li> </ul>

The following table describes the additional columns in the DataHub topic.

Previous additional column name	New additional column name	Data type	Description
dts_operat ion_flag	new_dts_sync_dts_ope ration_flag	String	<ul> <li>The operation type. Valid values:</li> <li>I: an INSERT operation</li> <li>D: a DELETE operation</li> <li>U: an UPDATE operation</li> </ul>
dts_instan ce_id	<pre>new_dts_sync_dts_ins tance_id</pre>	String	The server ID of the database. The value is set to null . To ensure database security, the actual value is not displayed.
dts_db_nam e	new_dts_sync_dts_db_ name	String	The name of the database.
dts_table_ name	new_dts_sync_dts_tab le_name	String	The name of the table.
dts_utc_ti mestamp	<pre>new_dts_sync_dts_utc _timestamp</pre>	String	The operation timestamp, in UTC. It is also the timestamp of the binary log file.
dts_before _flag	<pre>new_dts_sync_dts_bef ore_flag</pre>	String	Indicates whether the column values are pre-update values. Valid values: Y and N.
dts_after_ flag	<pre>new_dts_sync_dts_aft er_flag</pre>	String	Indicates whether the column values are post-update values. Valid values: Y and N.

# Additional information about the dts\_before\_flag and dts\_after\_flag fields

The values of the dts\_before\_flag and dts\_after\_flag fields in an incremental log entry vary with different operation types:

INSERT

For an INSERT operation, the column values are the newly inserted record values (post-update values). The value of the dts\_before\_flag field is N, and the value of the dts\_after\_flag field is Y.

dts_id	dts_name	dts_address	dts_record_id	dts_operation_flag	dts_instance_id	dts_db_name	dts_table_name	dts_utc_timestamp	dts_before_flag	dts_after_flag
10112	194		1574832919000000000	I.		dtstestdata	customer	1574832919	N	Y

• UPDATE

DTS generates two incremental log entries for an UPDATE operation. The two incremental log entries have the same values for the <code>dts\_record\_id</code>, <code>dts\_operation\_flag</code>, and <code>dts\_utc\_timestamp</code> fields.

The first log entry records the pre-update values. Therefore, the value of the dts\_before\_flag field is Y, and the value of the dts\_after\_flag field is N. The second log entry records the post-update values. Therefore, the value of the dts\_before\_flag field is N, and the value of the dts\_after\_flag field is Y.

dts_id	dts_name	dts_address	dts_record_id	dts_operation_flag	dts_instance_id	dts_db_name	dts_table_name	dts_utc_timestamp	dts_before_flag	dts_after_flag
10006	192		1574832130000000000	U		dtstestdata	customer	1574832130	Y	N
10006	100		1574832130000000000	U		dtstestdata	customer	1574832130	N	Y

#### • DELETE

For a DELETE operation, the column values are the deleted record values (pre-update values). The value of the dts\_before\_flag field is Y, and the value of the dts\_after\_flag field is N.

dts_id	dts_name	dts_address	dts_record_id	dts_operation_flag	dts_instance_id	dts_db_name	dts_table_name	dts_utc_timestamp	dts_before_flag	dts_after_flag
10009	100	1010	1574832919000000000	D		dtstestdata	customer	1574832919	Y	N

# What to do next

After you configure the data synchronization task, you can use Realtime Compute for Apache Flink to analyze the data that is synchronized to the DataHub instance. For more information, see What is Alibaba Cloud Realtime Compute for Apache Flink?

# 8.5. Synchronize data from a PolarDB for MySQL cluster to an Elasticsearch cluster

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

# Prerequisites

- An Elasticsearch cluster of version 5.5, 5.6, 6.3, 6.7, or 7.4 is created. For more information, see Create an Elasticsearch cluster.
- The binary logging feature is enabled for the 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 loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- DTS does not synchronize data definition language (DDL) operations. If a DDL operation is performed on a 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 cluster, 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 mapping of the table in the Elasticsearch cluster, perform DDL operations in the PolarDB for MySQL cluster, and then pause and start the data synchronization task.

#### SQL operations that can be synchronized

INSERT, DELETE, and UPDATE

#### Data type mappings

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

#### Procedure

(Optional)

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

(?) Note On the buy page, set Source Instance to PolarDB, Destination Instance to Elasticsearch, 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.

1.Configure Source and Destinatio	on 2.Select Objects to Synchronize		A.Precheck
Synchronization Task Name:	PolarDB MySQL_TO_Elasticsearch		
Source Instance Details			
Instance Type:	PolarDB Instance	*	
Instance Region:	China (Hangzhou)		
* PolarDB Instance ID:	nc hn		
	pc-op.	•	
* Database Account:	dtstest		
Database Password:	******	<4>	
Destination Instance Dataile			
Descination Instance Decails			
Instance Type:	Flasticsearch		
Instance Region:	China (Hangzhou)		
* Elasticsearch	es-cn-	-	
* Database Account:	dtstest		
Database Password:	•••••	<b>₫</b> >	
			Cancel Set Whitelist and Next

Section	Parameter	Description
N/A	Synchronizat ion Task Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.
	lnstance Type	The value of this parameter is set to <b>PolarDB Instance</b> and cannot be changed.
Source Instance Details	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.
	PolarDB Instance ID	The ID of the source PolarDB for MySQL cluster.
		The database account of the PolarDB for MySQL cluster.
	Dat abase Account	<b>Note</b> The account must have read permissions on the source database.
	Dat abase Password	The password of the database account.
	lnstance Type	This parameter is set to <b>Elasticsearch</b> and cannot be changed.
Destination	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.
Instance	Elasticsearch	The ID of the destination Elasticsearch cluster.
Details	Dat abase Account	The account that is used to connect to the Elasticsearch cluster. The default account is elastic.
	Dat abase Password	The password of the database account.

7.

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

1 Configure Source and Destination	2 Coloct Objects to Supphropize		2 Advanced Cottings		4 Drochock
1.conligure source and Destination	2.Select Objects to Synchronize		5.Auvanceu Settings		4.FICUICCK
Synchronization Mode: One-Way Syn	ichronization				
Index Name: DatabseName_Tabl	eName 🔻				
Proccessing Mode In Existed Target Table:	check and Intercept $ igodot$ Ignore				
Available			Selected (To edit an object name	or its filter, hover ove	r the object and click
If you search globally, please ex	pand the Q		Eucy Learn more.		
+ 🚰 sys			dtstestdata (20hiects)	Q	
dtstestdata			dtstestdata_customer Sour	ce Table Na	
		>	dtstestdata_order Source T	able Na	
		-			
		<			
Select All			Select All		
Name batch change:      Normal Norma Normal Norm	) 🔍 Yes rou 🔿 Yes 💿 No ? p				
* Retry Time for Failed Connection	720 Minutes 🥐	)			
				Cancel Pr	evious Next Precheck
Darameter	Description				
Parameter	Description				
Index Name	<ul> <li>Table Name         If you select Table         Elasticsearch clust         the index name is         DatabaseName_         If you select Data     </li> </ul>	e Name er is the custom TableN baseNa	e, the name of the i e same as the name er. lame ame_TableName,	index that is e of the tab the name o	s created in the le. In this example, of the index that is
	this example, the	index na	ime is dtstestdata_	_customer.	

Parameter	Description
Parameter Processing Mode In Existed Target Table	<ul> <li>Pre-check and Intercept : checks whether the destination database contains indexes that have the same names as the source tables. If the destination database does not contain indexes that have the same names as the source tables, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started.</li> <li>Note If indexes in the destination database have the same names as the source tables, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see Rename an object to be synchronized.</li> <li>Ignore: skips the precheck for identical index names in the source and destination databases.</li> <li>Ignore: skips the precheck for identical index names in the source and destination databases.</li> <li>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 the the same as the same as the same</li></ul>
Select the objects to be synchronized	Select one or more objects from the <b>Available</b> section and click the > icon to move the objects to the <b>Selected</b> section. You can select tables or databases as the objects to be synchronized.
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.

Parameter	Description			
	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> </ul>			
Replicate Temporary Tables When DMS Performs DDI	<b>Note</b> If online DDL operations generate a large amount of data, the data synchronization task may be delayed.			
Operations	• <b>No</b> : DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.			
	<b>Note</b> If you select No, the tables in the destination database may be locked.			
Retry Time for Failed	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.			
Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.			

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

Edit Table	×					
<b>Information:</b> After you edit the table or column name in the source database, the corresponding table or column name in the destination database is also updated.						
* Index Name:	sys_config					
* Type Name:	sys_config					
Filter:	DTS supports the WHERE clause in SQL statements. Only data that meets the WHERE clause can be migrated to the destination					
Settings_routing:(	DYes 🖲 No 🥜					
_id value:	Bis id V Select an option.					
Select All Colu	column Imn Name Type param column param value					
✓ set_by	varchar(12 index 🗸 false 🗸 add param					
✓ set_time	timestamp index 🗸 false 🗸					
✓ value	varchar(12 index 🗸 false 🗸 add param					
✓ variable	varchar(12 index 🗸 false 🗸 add param					
	ОК					
Parameter	Description					
Index Name	For more information, see Terms.					
Type Name	<ul> <li>Warning</li> <li>The only type of special characters that an index name and type nam can contain is underscore (_).</li> <li>To synchronize multiple source tables with the same schema to a destination object, you must repeat this step to set the same index name and type name for the tables.</li> </ul>					

Description
Specify SQL conditions to filter data. Only the data records that meet the specified conditions are synchronized to the destination cluster. For more information, see Use SQL conditions to filter data.
Select whether to set partitions. If you select <b>Yes</b> , you must also specify the <b>partition key column</b> and <b>number of partitions</b> .
<ul> <li>Specify whether you want to store a document on a specified shard of the destination Elasticsearch cluster. For more information, see _routing.</li> <li>If you select Yes, you can specify custom columns for routing.</li> <li>If you select No, the _id value is used for routing.</li> <li>Note If the version of the destination Elasticsearch cluster is 7.4, you must select No.</li> </ul>
<ul> <li>Primary key column Composite primary key fields are merged into one column.</li> <li>Business key If you select a business key, you must also specify the business key column.</li> </ul>
Select the <b>column parameter</b> and <b>parameter value</b> . For more information, see Mapping parameters in the Elasticsearch documentation.
<b>Note</b> DTS supports only the parameters that can be selected.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the 🕡 icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) <del>-</del>	Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscription Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1 > »

# Check the index and data

If the data synchronization task is in the **Synchronizing** state, you can connect to the Elasticsearch cluster by using the Elasticsearch-Head plug-in. Then, you can check whether the index is created and data is synchronized as expected. For more information, see Use Cerebro to access an Elasticsearch cluster.

**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.

Elasticsearch http://e	.public.elasticse	es-	cn-		集群健康	ŧ值: g	reen (52 of 52)
the set of a set of the set of th							
Read .							
A 44 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	and a supervision of						
200	_index	_type	_id	_score ▲	address	id	name
.kibana_1	dtstestdata_customer	customer	2077	1		2077	Pati
.monitoring-es-6-2019.08.28	dtstestdata	customer	2079			2079	Harman Arstall
.monitoring-es-6-2019.08.29	dtstestdata_customer	customer	2083	1		2083	Katima Malaka
.monitoring-es-6-2019.08.30	dtstestdata_customer	customer	2087	1		2087	Electron
.monitoring-es-6-2019.08.31	dtstestdata_custoned	dex : dtste	stdata_o	customer",		2088	Sel
.monitoring-es-6-2019.09.01	dtstestdata_customend	": "2077",	2102	1		2102	Isis
.monitoring-es-6-2019.09.02	dtstestdata_custon_ve	rsion	2111	1		2111	Aar
.monitoring-es-6-2019.09.03	dtstestdata_customero	ore": 1, uting" <sup>[1</sup> "20"	<b>-7</b> 416	1		2116	Rut
.monitoring-kibana-6-2019.08.28	dtstestdata custon.er	_source": {	2123	1		2123	Did
.monitoring-kibana-6-2019.08.29	dtstestdata customer	"address":	Ξ.,	1		2127	Sig
.monitoring-kibana-6-2019.08.30	dtstestdata customer	"id": 2077, "nomo": " D	al			2134	Ten
.monitoring-kibana-6-2019.08.31	dtstestdate custo }er	customer	2126			2126	location
.monitoring-kibana-6-2019.09.01	dtstestdata_ct_stomer	customer	2130	1		2130	Pok
.monitoring-kibana-6-2019.09.02	dtstestdata_customer	customer	2139	-		2139	Rec
.monitoring-kibana-6-2019.09.03	dtstestdata_customer		2157	1		2157	Bar
.security-6	dtstestdata_customer		2159	1		2159	Ald
dtstestdata_customer	dtstestdata_customer	customer	2165	1	and the second second	2165	Coc
dtstestdata_order	dtstestdata_customer	customer	2167	1		2167	Fiel
	dtstestdata_customer	customer	2168	1		2168	The
customer	dtstestdata_customer	customer	2180	1	-	2180	Ora
doc	dtstestdata_customer	customer	2185	1		2185	Gip
dtstestdata_order	dtstestdata_customer	customer	2187	1		2187	Erti

# 8.6. Synchronize data from a PolarDB for MySQL cluster to a self-managed Kafka cluster

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

# Prerequisites

- A Kafka cluster is created and the Kafka version is 0.10.1.0 to 2.7.0.
- The binary logging feature is enabled for the PolarDB for MySQL cluster. For more information, see Enable binary logging.

#### Precautions

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

# Limits

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

**?** Note If a source table is renamed during data synchronization but the new table name is not included in the selected objects, DTS does not synchronize the data of the table to the destination Kafka cluster. To synchronize the data of the renamed table, you must add the table to the selected objects of the task. For more information, see Add an object to a data synchronization task.

# Procedure

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

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

- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click **Data Synchronization**.
- 4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.
- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

#### Dat a Transmission Service

#### Data Synchronization Synchronize d ata to or from an Apsara PolarDB fo r MySQL cluster

Synchronization Task Name:	PolarDB MySQL_To_Kafka				
Source Instance Details					
Instance Type:	PolarDB Instance	~			
Instance Region:	China (Hangzhou)				
* PolarDB Instance ID:	pc-	•			
* Database Account:	dtstest				
* Database Password:	•••••	<			
Destination Instance Details					
Instance Type:	User-Created Database in ECS Instance	~			
Instance Region:	China (Shanghai)				
* ECS Instance ID:	i-man and any list of	•			
Database Type:	Kafka	~			
* Port Number:	9092				
Database Account:			Optional		
Database Password:		<	Optional		
* Topic:	polardb_test	~	Get Toplic list		
	Click Get Topic List and then select the specific topic.				
* Kafka Version	0.10	~			
* Encryption:	● Non-encrypted O SCRAM-SHA-256				
				Cancel	Set Whitelist and Next

Section	Parameter	Description
N/A	Synchronizat ion Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
	lnstance Type	This parameter is set to <b>PolarDB Instance</b> and cannot be changed.
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.
	PolarDB Instance ID	Select the ID of the PolarDB for MySQL cluster.
Source Instance Details	Dat abase Account	Enter the database account of the PolarDB for MySQL cluster. The account must have the read permissions on the objects to be synchronized.
	Database Password	Enter the password of the database account.

Section	Parameter	Description				
	lnst ance T ype	Select an instance type based on the deployment of the Kafka cluster. In this example, select <b>User-Created Database in ECS Instance</b> .				
		<b>Note</b> If you select other instance types, you must deploy the network environment for the Kafka cluster. For more information, see <b>Preparation overview</b> .				
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.				
	ECS Instance ID	Select the ID of the Elastic Compute Service (ECS) instance on which the Kafka cluster is deployed.				
	Database Type	Select Kafka.				
Destination Instance Details	Port Number	Enter the service port number of the Kafka cluster. The default port number is 9092.				
	Dat abase Account	Enter the username that is used to log on to the Kafka cluster. If no authentication is enabled for the Kafka cluster, you do not need to enter the username.				
	Dat abase Password	Enter the password that corresponds to the username. If no authentication is enabled for the Kafka cluster, you do not need to enter the password.				
	Торіс	Click <b>Get Topic List</b> , and select a topic name from the drop-down list.				
	Kafka Version	Select the version of the destination Kafka cluster.				
	Encryption	Select <b>Non-encrypted</b> or <b>SCRAM-SHA-256</b> based on your business and security requirements.				

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

#### 8. Select the objects to synchronize.

1.Select Source and Destination		> 4.Precheck
Synchronization Mode:One-Way Synchronization Data format delivered to Kafka:		
Synchronize to Kafka Partition Policy® Post all to Partition 0 ⑦ ○ Delivered different parts by hash value of libra ○ Delivered different parts by hash value of prin Note: After the data synchronization operation is synchronization will fail	ary name + table n nary key officially started, c	name do not modify the number of partitions of the target topic, otherwise the
Available Expand the tree before you perform a glo Q C thw Tables C hw02 C the tree before you perform a glo Q C the tree before	> <	Selected (To edit an object name or its filter, hover over the object and dick Edit.) Learn more.
Select All *Rename Databases and Tables: © Do Not Change Database and	d Table Names	Select All O Change Database and Table Names
* Retry Time for Failed Connection 720 Minutes (	0	
		Cancel Previous Next

Parameter	Description
Data Format in Kafka	The data that is synchronized to the Kafka cluster is stored in the Avro or Canal JSON format. For more information, see Data formats of a Kafka cluster.
Policy for Shipping Data to Kafka Partitions	The policy used to synchronize data to Kafka partitions. Select a policy based on your business requirements. For more information, see Specify the policy for synchronizing data to Kafka partitions.
	Select one or more tables from the <b>Available</b> section and click the <b>&gt;</b> icon to add the tables to the <b>Selected</b> section.
Select the objects to synchronize	<b>Note</b> DTS maps the table names to the topic name that you select in Step 6. If you want to rename the topic, you can use the object name mapping feature. For more information, see <b>Rename an object to be synchronized</b> .

Parameter	Description
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.
Retry Time for Failed	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.
and Tables Retry Time for Failed Connections	By default, if DTS fails to connect to the source or destination database, DT retries within the next 720 minutes (12 hours). You can specify the retry tim based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.

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

#### 10. Configure initial synchronization.

1.Select Source and Destination Instances $ig>$			4.Precheck					
Initial Synchronization: Imitial Schema Synchronization Imitial Full Data Synchronization Note: Trigger synchronization is not supported, please Reference Document Filter options: Imitial Synchronization phase								
Cancel Previous Save Precheck								
Setting	Description							
Initial Synchronization	Select both <b>Initial Schema Synchronization</b> and <b>Initial Full Data</b> <b>Synchronization</b> . DTS synchronizes the schemas and historical data of the required objects and then synchronizes incremental data.							
Filter options	<b>Ignore DDL in incremental synchronization phase</b> is selected by default. In this case, DTS does not synchronize DDL operations that are performed on the source database during incremental data synchronization.							

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.
- 12. Close the **Precheck** dialog box after the following message is displayed: **Precheck Passed**. Then, the data synchronization task starts.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscription Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < <b>1</b> > »

# 8.7. Synchronize data from a PolarDB for MySQL cluster to an AnalyticDB for MySQL cluster

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

# Prerequisites

- An cluster is created. For more information, see Create an cluster.
- The destination cluster has sufficient storage space.
- The binary logging feature is enabled for the 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 loads of the database servers. 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 impact of data synchronization on 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 utilization 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 data definition language (DDL) operations on the required objects during data synchronization. Otherwise, data may fail to be synchronized.
- Due to the limits of , if the disk space usage of the nodes in an cluster reaches 80%, the cluster is locked. We recommend that you estimate the required disk space based on the objects that you want to synchronize. You must ensure that the destination cluster has sufficient storage space.
- Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to be synchronized.

# SQL operations that can be synchronized

- DDL operations: CREATE TABLE, DROP TABLE, RENAME TABLE, TRUNCATE TABLE, ADD COLUMN, DROP COLUMN, and MODIFY 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 permissions
PolarDB for MySQL	The read permissions on the objects to be synchronized
	The read and write permissions on the objects to be synchronized

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

# Data type mappings

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

# Procedure

1. Purchase a data synchronization instance.

On the buy page, set Source Instance to PolarDB, 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.

1.Configure Source and Destinati	on Instances 2.Author	ize AnalyticDB Account 💦 🔪		3.Select Objects to Synchronize	$\geq$	4.Precheck
			_			
Synchronization Task N	ame: PolarDB MySQL_TO_ADB MySQ	L				
Source Instance Details						
Instance	TVDE: PolarDB Instance		•			
Instance Re	gion: Singapore					
* PolarDB Instance	e ID: pc-gs		•			
* Database Acco	ount: dtstest					
* Database Passy	vord:		க			
Database i assi			4.			
Destination Instance Details						
Instance T	Type: AnalyticDB					
Instance Re	gion: Singapore					
* Datal	base: am-gs		•			
* Database Acco	ount: dtstest					
* Database Passe	word:		<b>(</b> )			
Database Passi			42			
						Cancel Set Whitelist and Next
Section	Parameter	Description				
N/A	Synchronizatio n Task Name	DTS automatica you specify an not need to use	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.			
	Instance Type	This parameter is set to <b>PolarDB Instance</b> and cannot be changed.				
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.				
Source Instance	PolarDB Instance ID	Select the ID of the source PolarDB cluster.				
Details	Dat abase Account	Enter the database account of the source PolarDB cluster. For information about the permissions that are required for the account, see Permissions required for database accounts.				
	Database Password	Enter the passv	word	l of the database acc	count.	
	Instance Type	This parameter	r is s	et to <b>AnalyticDB</b> and	d canno	t be changed.
	Instance Region	The destination cannot change	n reg the	jion that you selected value of this parame	d on th ter.	e buy page. You

Section	Parameter	Description
Destination Instance Details	Version	Select 3.0.
	Database	Select the ID of the destination cluster.
	Database Account	Enter the database account of the destination cluster. For information about the permissions that are required for the account, see Permissions required for database accounts.
	Dat abase Password	Enter the password of the 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 PolarDB for MySQL cluster and the cluster. This ensures that DTS servers can connect to the source and destination clusters.

8. Select the synchronization policy and the objects to be synchronized.

1.Select Source and Destina	tion 🔰 2.Authorize AnalyticDB	Account	3.Select Object to Be	4.Precheck
Initial Synchronization: 🔽	Initial Schema Synchronization	Initial Full Data Synch	ronization	
Note: do not clean up the inci cleans up the log too early th	remental data log generated by the sour	ce database after the	DTS task is started when the DTS full task	is running. If the source database
Proccessing Mode In	e of a incremental task may fair			
Existed Target Tables	Pre-check and Intercept O Ignore			
Synchronization Type:	′es   ● No Insert    ✓ Update     ✓ Delete	Alter Table	✓ Truncate Table	
	Create Table 🗹 Drop Table			
Available			Selected (To edit an object name or its Edit.) Learn more.	filter, hover over the object and click
Expand the tree before you p	perform a glol 🔰 🍳			0
🕀 🦢recycle_bin			📑 dtstestdata	2
⊞ 💼 asd ⊞ 💼 chw02				
🕀 🦢 dts		>		
Itstest0512_jzhz_00 Itstest123	01_ext_0001	,		
🗄 📂 dtstestdata1		<		
Select All			Select All	
*Rename Databases and Tables	: 💿 Do Not Change Databa	e and Table Names	O Change Database and Table Names	1
*Source table DMS_ ONLINE_ D want to copy the temporary tabl the target database during DDL:	oyou 💿 Yes 🔿 No 🧭 eto			
* Retry Time for Failed Connecti	ion 720 Minute	ıs 🕜		
			Cancel Pr	evious Next Precheck
Parameter	Description			
nitial Synchronization	You must select bot Synchronization in schemas and data c destination cluster.	h <b>Initial Sch</b> most cases. f the required The schemas	<b>ema Synchronization</b> After the precheck, DTS d objects from the sourc and data are the basis f	and <b>Initial Full Data</b> synchronizes the e cluster to the or subsequent

Parameter	Description			
Processing Mode In Existed Target Table	<ul> <li>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.</li> <li>Note You can use the object name mapping feature to rename the tables that are synchronized to the destination database. You can use this feature if the source and destination database contain identical table names and the tables in the destination database cannot be deleted or renamed. For more information, see Rename an object to be synchronized.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>Ignore: skips the precheck for identical table names in the source and destination databases.</li> <li>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.</li> <li>If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only specific columns are synchronized or the data synchronization task fails.</li> </ul>			
Merge Multi Tables	<ul> <li>If you select Yes, DTS adds thedts_data_source column to each table to record data sources. In this case, DDL operations cannot be synchronized.</li> <li>No is selected by default. In this case, DDL operations can be synchronized.</li> <li>Note You can merge the data source columns based on tasks rather than tables. To merge only the data source columns of specific tables, you can create two data synchronization tasks.</li> </ul>			
Synchronization Type	Select the types of operations that you want to synchronize based on your business requirements. All operation types are selected by default. For more information, see SQL operations that can be synchronized.			

Parameter	Description				
Select the objects to be synchronized	Select one or more objects from the <b>Available</b> section and click the icon to move the objects to the <b>Selected</b> section. You can select tables or databases as the objects to be synchronized.				
	<ul> <li>Note</li> <li>If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.</li> <li>If you select a table as the object to be synchronized, only the ADD COLUMN operations that are performed on the table are synchronized to the destination database.</li> </ul>				
	<ul> <li>After an object is synchronized to the destination cluster, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination cluster. For more information, see Rename an object to be synchronized.</li> </ul>				
Rename Dat <i>a</i> bases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.				
	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> </ul>				
Replicate Temporary Tables When DMS	<b>Note</b> If online DDL operations generate a large amount of data, the data synchronization task may be delayed.				
Performs DDL Operations	• <b>No</b> : DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.				
	<b>Note</b> If you select No, the tables in the destination database may be locked.				

Parameter	Description
Retry Time for Failed Connections	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

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

	Destination Instances 2.Au		3.Select Objects to	Synchronize	
AnalyticDB Table Group	AnalyticDB Table Name	Type(All) 👻	Primary Key Column	Distribution Column	Definition Status(All) 👻
dtstestdata	customer	Partitioned 1 🔻	id	id 🔻	Defined
dtstestdata	order	Partitioned 1 V	orderid	orderid •	Defined
Set All to Partitioned Table	Set All to Dimension Table Enter a table na	ame. Search		Total: 2 item(s), Per Page: 20	) • item(s) « < 1 > »
				Cancel	Previous Save Precheck

Onte After you select Initial Schema Synchronization, you must specify the type, primary key column, and partition key column for the tables that you want to synchronize to. 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, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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



# 8.8. Synchronize data from a 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 that can process petabytes of data. This topic describes how to synchronize data from a PolarDB for MySQL cluster to an AnalyticDB for PostgreSQL instance by using Data Transmission Service (DTS). The data synchronization feature is applicable to scenarios such as ad hoc query and analysis, extract, transform, and load (ETL) operations, and data visualization.

# Prerequisites

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

# Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After initial full data synchronization is complete, 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.
- DTS does not synchronize the following types of data: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID\_SNAPSHOT.
- Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to 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.
#### SQL operations that can be synchronized

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

Onte 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

PolarDB for MySQL	
Database	Schema
Table	Table

#### Procedure

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

On the buy page, set Source Instance to PolarDB, 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.

Cancel

Set Whitelist and Next

1.Select Source and Destination	nstances for 2.Select	Object to Be Synchronized	$\rightarrow$	3.Precheck
Synchronization Task Name:	polardb_to_adbpg			
Source Instance Details				
Instance Type:	PolarDB Instance	~		
Instance Region:	China (Hangzhou)			
* PolarDB Instance ID:				
Fourbo Insurice ID.	pc-	•		
* Database Account:	dtstest			
* Database Dassword				
Database Password.	•••••	4/		
Destination Instance Details				
Instance Type:	AnalyticDB for PostgreSQL			
Instance Region:	China (Hangzhou)			
* Instance ID:	gp-	•		
* Database Name:	dtstestdata			
* Database Account:	dtstest			
* Database Password:	*****	<b>∢</b> >		

Section Parameter Description Synchronizat DTS automatically generates a task name. We recommend that you N/A ion Task specify an informative name for easy identification. You do not need Name to use a unique task name. Instance This parameter is set to **PolarDB Instance** and cannot be changed. Туре Instance The source region that you selected on the buy page. You cannot Region change the value of this parameter. PolarDB Select the ID of the PolarDB for MySQL cluster. Instance ID Enter the database account of the PolarDB for MySQL cluster. Source **?** Note The database account must have the read Instance Database permissions on the objects to be synchronized. Details Account Database Enter the password of the database account. Password

Section	Parameter	Description			
	lnstance Type	This parameter is set to <b>AnalyticDB for PostgreSQL</b> and cannot be changed.			
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.			
	Instance ID	Select the ID of the instance.			
Database Name	Database Name	Enter the name of the destination database in the instance.			
Instance Details	Destination Instance Details Database Account	Enter the <b>initial account</b> of the instance. For more information, see <b>Create a database account</b> .			
		<b>Note</b> You can also enter an account that has the RDS_SUPERUSER permission. For more information, see Manage users and permissions.			
	Dat abase Password	Enter the password of the 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 PolarDB for MySQL cluster and the instance. This ensures that DTS servers can connect to the source cluster and the destination instance.

#### 8. Select the synchronization policy and the objects to be synchronized.

,	o financia di di di	<b>.</b>			
<ul> <li>Note: do not clean up the incremental d cleans up the log too early, the DTS incr</li> </ul>	lata log generated by remental task may fai	the source dat I	abase after the D	TS task is started when the DTS full task is running. If the source d	atabase
Droccossing Mode In					
Existed Target Table:   Pre-check a	and Intercept $ \bigcirc $ Clea	ar Target Table	○ Ignore		
Synchronization Type: 🗹 Insert	✔ Update	Delete	✓ Alter Table		
Available				Selected (To edit an object name or its filter, hover over the objectit.) Learn more.	ct and cli
Expand the tree before you perform a g	glol 🛛 🔍				
🕀 📴recycle_bin					
🕀 🗁 asd				dtstestdata	
🖃 📴 chw02					
H dtstest0512 jzhz 0001 ext 0	0001		>		
			1		
🗄 🦢 dtstestdata 1			<		
🕀 🏪 sys					
Select All					
				Select All	
Rename Databases and Tables:	Do Not Change	Database and	Table Names	<ul> <li>Change Database and Table Names</li> </ul>	
Source table DMS_ ONLINE_ Do you	◉ Yes 🔿 No	0			
he target database during DDL:		Minutes (?	)		
the target database during DDL: Retry Time for Failed Connection	720				

Setting	Parameter	Description
	Initial Synchronization	You must select both <b>Initial Schema Synchronization</b> and <b>Initial Full Data Synchronization</b> 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.

Setting	Parameter	Description		
Select the synchronizatio n policy	Processing Mode of Conflicting Tables	<ul> <li>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.     <li>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 tables to one table, you can select this mode.     </li> </li></ul>		
	Synchronization Type	<ul> <li>Select the types of operations that you want to synchronize based on your business requirements.</li> <li>Insert</li> <li>Update</li> <li>Delete</li> <li>AlterTable</li> </ul>		
Select the objects to be synchronized	N/A	<ul> <li>Select one or more tables from the Available section and click the icon to move the tables to the</li> <li>Selected section.</li> <li>? Note <ul> <li>You can select only tables as the objects to be synchronized.</li> <li>You can use the object name mapping feature to change the names of the columns that are synchronized to the destination database. For more information, see Rename an object to be synchronized.</li> </ul> </li> </ul>		
Rename Databases and Tables	N/A	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.		

Setting	Parameter	Description		
Replicate Temporary Tables When DMS Performs DDL Operations	N/A	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> <li>Yote If online DDL operations generate a large amount of data, the data synchronization task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> </ul>		
Retry Time for Failed Connections	N/A	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.		
		you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.		

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

1.Configure So	urce and Destination I	nstances in	2.Select Objects to Synchronize	3.Precheck
Schema	Table	Primary Key Column	Distribution Column	Definition Status(All) 👻
dtstestdata	customer	id	id 💌	Defined
dtstestdata	order	orderid	orderid <b>v</b>	Defined
dts.migration.messag	e.greenplu Search		То	otal: 2 item(s), Per Page: 20 $\bullet$ item(s)
				Cancel Previous Save Precheck

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 Define constraints and Define table distribution.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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.

	Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
	0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
	Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1	> >>

# 8.9. Synchronize data from a PolarDB for MySQL cluster to a PolarDB-X instance

Distributed Relational Database Service (PolarDB-X 1.0) is developed by Alibaba Group to address the bottleneck of single-host database services. DRDS is compatible with the MySQL protocol and syntax, and supports automatic sharding, online smooth scaling, auto scaling, and transparent read/write splitting. DRDS provides O&M capabilities throughout the lifecycle of databases. This topic describes how to synchronize data from a PolarDB for MySQL cluster to a PolarDB-X 1.0 instance by using Data Transmission Service (DTS).

#### Prerequisites

- The binary logging feature is enabled for the PolarDB for MySQL cluster. For more information, see Enable binary logging.
- The tables to be synchronized from the source database contain primary keys.

- The destination database has sufficient storage space.
- A PolarDB-X instance is created. For more information, see Create a instance and Create a database.

**?** Note When you create an instance, you must select RDS MySQL as the storage type.

#### Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination cluster. After initial full data synchronization, the tablespace of the destination cluster is larger than that of the source database. You must make sure that the destination database has sufficient storage space.
- DTS does not synchronize schemas from an ApsaraDB RDS for MySQL instance to a PolarDB-X 1.0 instance. Before you configure a data synchronization task, you must create databases and tables in the destination instance.

#### Limits

- You can select only tables as the objects to be synchronized.
- DTS does not synchronize the following types of data: BIT, VARBIT, GEOMETRY, ARRAY, UUID, TSQUERY, TSVECTOR, and TXID\_SNAPSHOT.
- Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to 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.

### SQL operations that can be synchronized

INSERT, UPDATE, and DELETE

#### Permissions required for database accounts

Database	Required permissions
PolarDB for MySQL	The database account must have the SELECT permission on the objects to be synchronized, the REPLICATION CLIENT permission, the REPLICATION SLAVE permission, and the SHOW VIEW permission.
PolarDB-X 1.0	DTS automatically creates a database account and grants permissions to the account. You do not need to specify the database account.

#### Supported synchronization topologies

One-way one-to-one synchronization

• One-way many-to-one synchronization

#### Supported synchronization topologies

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

#### Before you begin

When you synchronize data from a PolarDB for MySQL cluster to a PolarDB-X 1.0 instance, note that DTS does not support **initial schema synchronization**. Therefore, you must create databases and tables in the destination instance based on the schemas of the objects in the source PolarDB for MySQL cluster. For more information, see Create a DRDS database and Create a DRDS table.

**?** Note During initial schema synchronization, DTS synchronizes the schemas of the required objects from the source database to the destination database.

#### Procedure

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

On the buy page, set Source Instance to PolarDB, set Target Instance to PolarDB-X, 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 cluster and the destination instance.

1.Select Source and Destination Instances for	2.Select Object to Be Synchronized	$\rightarrow$	3.Advanced Settings	· > · · · ·	ł.Precheck
Synchronization Task Name: polardb to d	rds				
pourus_to_u					
Source Instance Details					
Instance Type: PolarDB Insta	ance 🗸				
Instance Region: China (Hangzh	ou)				
* PolarDB Instance ID: pc-					
* Database Account: dtstest					
* Database Password:	4>				
Destination Instance Details					
Instance Type: DRDS Instance	2				
Instance Region: China (Hangzh	ou)				
* DRDS Instance ID: drds	· ·				
				Cancel	Set Whitelist and Next

Section	Parameter	Description
N/A	Synchronizatio n Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.
	Instance Type	Select PolarDB Instance.
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.
Source Instance	PolarDB Instance ID	Select the ID of the source PolarDB cluster.
Details	Database Account	Enter the database account of the source PolarDB cluster. For information about the permissions that are required for the account, see Permissions required for database accounts.
	Database Password	Enter the password of the database account.
	Instance Type	This parameter is set to <b>DRDS</b> Instance and cannot be changed.
Destination Instance	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.
Details	DRDS Instance ID	Select the ID of the destination DRDS instance.

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

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

8. Select the synchronization policy and the objects to be synchronized.

1.Select Source and Destination	2.Select Object to Be Synchronized	3.Advanced Settings > 4.Precheck
Synchronization Mode:	One-Way Synchronization	
Available		Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.
Expand the tree before y	ou perform a glo Q	Carlo (Learn Note) Q Q ⇒ chw (10bjects) ■ tw02 U002 Object name distination di select the col migrate. To ensure co we will Conve database nam lowercase by you need to k uppercase, pi them manual
Select All		Select All
*Rename Databases and Ta	les: <ul> <li>Do Not Change Database and Table National Statement</li> </ul>	mes O Change Database and Table Names
		Cancel Previous Next
Setting	Description	
Select the objects to be synchronized	Select one or more tables from move the tables to the Select <b>Oremove Oremove <b>Oremove Oremove Oremove <b>Oremove Oremove Oremove Oremove <b>Oremove</b></b></b></b>	m the Available section and click the > icon to ted section. tables as the objects to be synchronized. object is synchronized to the destination of the object remains unchanged. You can use apping feature to change the names of the uchronized to the destination instance. For more name an object to be synchronized.
Rename Databases and Tables	You can use the object name synchronized to the destination mapping.	mapping feature to rename the objects that are on instance. For more information, see Object name

Setting	Description
Retry Time for	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

#### 9. Click Next.

10. Specify whether you want to perform initial full data synchronization.

	2.Select Object to Be Synchronized	3.Advanced Settings	4.Precheck
Initial Synchronization: Initial Full Data Synchron	nization		
			Cancel Previous Save Precheck

**?** Note During initial full data synchronization, DTS synchronizes the historical data of the required objects from the source database to the destination database. If you do not select Initial Full Data Synchronization, DTS does not synchronize the historical data.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the 🕧 icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscription Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < <b>1</b> > »

# 9.Synchronize data to or from a PolarDB for Oracle cluster

## 9.1. Configure one-way data synchronization between PolarDB for Oracle clusters

PolarDB for Oracle is a next-generation cloud-native relational database service that is developed by Alibaba Cloud. PolarDB for Oracle is highly compatible with the Oracle database engine. This topic describes how to configure one-way data synchronization between PolarDB for Oracle clusters by using Data Transmission Service (DTS).

#### Prerequisites

- The source PolarDB for Oracle cluster uses the latest version. For more information, see Version Management.
- The tables to be synchronized contain primary keys or UNIQUE NOT NULL indexes.
- The value of the wal\_level parameter is set to *logical* for the source PolarDB for Oracle cluster. This setting ensures that logical decoding is supported in write-ahead logging (WAL). For more information, see Configure cluster parameters.

#### Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization 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 <u>dts\_postgres\_heartbeat</u> to the source database. The following figure shows the schema of the heartbeat table.

exect	ite(F8) Row Details	Plan(F7) Forma	at(F9)					
1 se	<pre>lect * from "dtstes</pre>	t"."dts_postgres_hea	artbeat"					
Mess	iges Results1						Cross Database SQL Q	uery
	SLOT_NAME	REVICE_TIME	▼ REVICE_LSN	▼ FLUSHED_LSN	UPDATE_TIME	v	DTS_SERVICE_TIME	*
1	w8i	1585104942560	0/44		2020-03-25 10:55:47.585187+08		1585104947579	

• If you select a schema as the object to be synchronized and create a table in the schema or run the RENAME command to rename the table, you must run the ALTER TABLE schema.table REPLICA IDENT ITY FULL; command before you write data to the table.

**Note** Replace the schema and table in the preceding sample command with the actual schema name and table name.

#### SQL operations that can be synchronized

Operation type	SQL statements
DML	INSERT, UPDATE, and DELETE
	<ul> <li>CREATE TABLE and DROP TABLE</li> <li>ALTER TABLE, including RENAME TABLE, ADD COLUMN, ADD COLUMN DEFAULT, ALTER COLUMN TYPE, DROP COLUMN, ADD CONSTRAINT, ADD CONSTRAINT CHECK, and ALTER COLUMN DROP DEFAULT</li> <li>CREATE INDEX ON TABLE</li> <li>? Note Data definition language (DDL) statements are not</li> </ul>
	<ul> <li>synchronized in the following scenarios:</li> <li>Additional information such as CASCADE and RESTRICT in DDL statements is not synchronized.</li> <li>If a transaction contains both DML and DDL statements, the DDL</li> </ul>
DDL	<ul> <li>statements are not synchronized.</li> <li>If only some DDL statements of a transaction are included in the objects of the data synchronization task, the DDL statements are not synchronized.</li> </ul>
	<ul> <li>If a DDL statement is executed in a session that is created by running the SET session_replication_role = replica command, the DDL statement is not synchronized.</li> </ul>
	<ul> <li>If no schema is defined in a DDL statement, the DDL statement is not synchronized. In this case, the public schema is specified in the SHOW search_path command.</li> </ul>
	• If a DDL statement contains IF NOT EXISTS, the DDL statement is not synchronized.

#### Procedure

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

**Note** On the buy page, set both Source Instance and Destination 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.

1.Configure Source and Destinatio	on 2.Select Objects to Synchronize		
Synchronization Task Name:	PolarDB O		
-,			
Source Instance Details			
Instance Type:	DelayDD Instance	-	
instance rype.	PolarDB Instance	•	
Instance Region:	China (Hangzhou)		
* PolarDB Instance ID:	pc-bp	-	
* Database Name:	dtstestdata		
* Database Account:	dtraupar		
Ditabase Accounts	utsownei		
* Database Password:	•••••	<>	
Destination Instance Details			
Instance Type:	PolarDB		
Instance Pagion:	(hina (Hangabau)		
Instance Region.	China (Hangzhou)		
* PolarDB Instance ID:	pc-bp	-	
* Database Name:	dtstestdata		
* Database Account:	dtstest		
* Database Password:	•••••	₫>	
			Cancel Set Whitelist and Next

Section	Parameter	Description
N/A	Synchroniz at ion Task Name	DTS automatically generates a task name. We recommend that you specify an informative name to identify the task. You do not need to use a unique task name.
	lnstance Type	The value of this parameter is set to <b>PolarDB Instance</b> and cannot be changed.
Source Instance Details	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.
	PolarDB Instance ID	Select the ID of the source PolarDB for Oracle cluster.
	Database Name	Enter the name of the source database.
	Dat abase Account	Enter the privileged account of the source PolarDB for Oracle cluster. For more information about how to create a privileged database account, see Create database accounts.

Section	Parameter	Description		
	Dat abase Password	Enter the password of the database account.		
	lnstance Type	The value of this parameter is set to <b>PolarDB</b> and cannot be changed.		
-	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.		
	PolarDB Instance ID	Select the ID of the destination PolarDB for Oracle cluster.		
Destination	Database Name	Enter the name of the destination database.		
Details	Database Account	Enter the database account of the destination PolarDB for Oracle cluster. The account must have the permissions of the <b>database owner</b> .		
		<b>Notice</b> You can specify the <b>database owner</b> when you create a database.		
	Databasa			
	Password	Enter the password of the database account.		

7.

8. Select the processing mode of conflicting tables and the objects to be synchronized.

Available			Selected (To edit an object name	e or its filter, hover over the object and c	lick Hover ow
Expand the tree before you per anydata aq\$_agent aq\$_agent aq\$_reg_info aq\$_reg_info aq\$_reg_info ad\$ms_alert ad\$ms_aq ad\$ms_aqdm ad\$ms_aqdm ad\$ms_lob ad\$ms_lob ad\$ms_lob ad\$ms_lock ad\$ms_mview	form a glo I Q	× ×	dbms_job		biject an appears, object na destinatic select the migrate. To ensure we will co database lowercase uppercase them mar
doms_output		-			
Select All			Select All		
*Rename Databases and Tables	Do Not Change Date	tabace and Table Names	Change Database and Table	Namec	

Setting	Description
	• <b>Pre-check and Intercept</b> : 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.
	<b>Note</b> You can use the object name mapping feature to rename the tables that are synchronized to the destination database. You can use this feature if the source and destination databases contain identical table names and the tables in the destination database cannot be deleted or renamed. For more information, see Rename an object to be synchronized.
Select the processing mode of conflicting	• <b>Ignore</b> : skips the precheck for identical table names in the source and destination databases.
tables	<ul> <li>Warning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks.</li> <li>During initial data synchronization, DTS does not synchronize the data records that have the same primary keys as the data records in the destination database. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization.</li> <li>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.</li> </ul>
	Select one or more objects from the <b>Available</b> section and click the > icon
	You can select tables or databases as the objects to be synchronized.
Select the objects to be synchronized	<ul> <li>Note</li> <li>If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination cluster, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination cluster. For more information, see Rename an object to be synchronized.</li> </ul>

Setting	Description
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.
Retry Time for Failed Connections	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

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

#### 10. Select the initial synchronization types.

1.Configure Source and Destination	2.Select Objects to Synchronize     3.Advanced Settings     4.Precheck
Initial Synchronization: 🗹 Initia	I Schema Synchronization 🔽 Initial Full Data Synchronization Note: Trigger synchronization is not supported, please Reference Document
	Cancel Previous Save Precheck
Initial synchronization type	Description
Initial schema synchronization	DTS synchronizes the schemas of the required objects to the destination PolarDB for Oracle cluster. DTS supports initial schema synchronization for the following types of objects: table, view, synonym, trigger, stored procedure, function, package, and user-defined type.
	<b>Notice</b> 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 <b>Configure a data</b> synchronization task for a source database that contains a trigger.
	PolarDB for Oracle cluster to the destination PolarDB for Oracle cluster.
Initial full data synchronization	<b>Notice</b> During 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, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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



# 9.2. Synchronize data from a PolarDB for Oracle cluster to a self-managed Kafka cluster

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

#### Prerequisites

- The source PolarDB for Oracle cluster uses the latest version. For more information, see Version Management.
- The tables to be synchronized contain primary keys or UNIQUE NOT NULL indexes.
- The value of the wal\_level parameter is set to *logical* for the source PolarDB for Oracle cluster. This setting ensures that logical decoding is supported in write-ahead logging (WAL). For more information, see Configure cluster parameters.

#### Precautions

- In this scenario, DTS supports only and are not supported. incremental data synchronizationSchema synchronizationf ull data synchronization
- A data synchronization task can synchronize data from only a single database. To synchronize data from multiple databases, you must create a data synchronization task for each database.
- To ensure that the latency 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.

exec	ute(F8) Row Details	; Pla	an(F7) Fo	ormat(F	9)							
1 se	<pre>lect * from "dtstes</pre>	t"."dt	s_postgres	heart	eat"							
Mess	Messages Results1 Cross Database SQL Query											
	SLOT_NAME	▼ R	EVICE_TIME	~	REVICE_LSN	Ŧ	FLUSHED_LSN	Ŧ	UPDATE_TIME	Ŧ	DTS_SERVICE_TIME	~
1	w8i	15	85104942560		0/44				2020-03-25 10:55:47.585187+08		1585104947579	

#### Procedure

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

(?) Note On the buy page, set Source Instance to PolarDB, Destination Instance to Kafka, 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.

Synchronization Task Name:	PolarDB-O_To_Kafka			
Source Instance Details				
Instance Type:	PolarDB Instance			
Instance Region:	China (Shanghai)			
PolarDB Instance ID:	pc			
* Database Name:	dtstestdata			
* Database Account:	dtstest			
Database Password:	<i>«</i>	]		
Destination Instance Details				
Instance Type:	User-Created Database in ECS Instance			
Instance Region:	China (Shanghai)			
* ECS Instance ID:	-			
Database Type:	Kafka 🔻			
* Port Number:	9092			
Database Account:		Is not necessary		
Database Password:	٩>	Is not necessary		
* Topic:	polardb_test	Get Toplic list		
* Kafka Version				
* Encryption:	Non-encrypted      SCRAM-SHA-256			
			Cancel	Set Whitelist and Next

Section	Parameter	Description
N/A	Synchronizat ion Task Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.
	lnstance Type	The value of this parameter is set to <b>PolarDB Instance</b> and cannot be changed.
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.
Source	PolarDB Instance ID	The ID of the source PolarDB for Oracle cluster.
Instance Details	Database Name	The name of the source database.
	Dat abase Account	A privileged account of the source PolarDB for Oracle cluster. For more information about how to create a privileged database account, see Create database accounts.
	Database Password	The password of the database account.
		The access method of the self-managed Kafka cluster. In this example, <b>User-Created Database with Public IP Address</b> is selected.
	Instance Type	<b>Note</b> If the self-managed Kafka cluster is connected over other methods, you must deploy the network environment for the Kafka cluster. For more information, see <b>Preparation overview</b> .
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.
	ECS Instance ID	The ID of the Elastic Compute Service (ECS) instance on which the Kafka cluster is deployed.
Dectination	Database Type	Select <b>Kafka</b> .
Instance	Port Number	The service port number of the Kafka cluster. Default value: 9092.
Derails	Dat abase Account	The username that is used to log on to the Kafka cluster. If no authentication is enabled for the Kafka cluster, you do not need to enter the username.
	Dat abase Password	The password that corresponds to the username. If no authentication is enabled for the Kafka cluster, you do not need to enter the password.
	Торіс	Click <b>Get Topic List</b> and select a topic name from the drop-down list.

Section	Parameter	Description
	Kafka version	The version of the self-managed Kafka cluster.
	Encryption	Select <b>Non-encrypted</b> or <b>SCRAM-SHA-256</b> based on your business and security requirements.

#### 7.

8. Select the objects to be synchronized.

1.Select Source and Destination		A.Precheck	
Synchronization Mode:One-W Data format delivered to Kafka: • • DT Synchronize to Kafka Partition Policy:• Pos O Del D Del Note: synchr	Vay Synchronization 5 Avro Canal Json t all to Partition 0 ivered different parts by hash value of library na ivered different parts by hash value of primary k After the data synchronization operation is offici ionization will fail	ame + table name key jaily started, do not modify the number of partitions of the target topic, otherwise the	
Available Expand the tree before you per chw Tables chw02 chw02 chw02 chw02	form a glo	Selected (To edit an object name or its filter, hover over the object and Edit.) Learn more.	dick
Select All		Select All	
*Rename Databases and Tables: * Retry Time for Failed Connection	Do Not Change Database and Tab     Z20 Minutes	O Change Database and Table Names	
		Cancel Previous	Next
Setting	Description		

Setting	Description
Select the objects to be synchronized	Select one or more tables from the <b>Available</b> section and click the <b>&gt;</b> icon to add the tables to the <b>Selected</b> section. You can select only tables as the objects to synchronize.
	<b>Note</b> DTS maps the table names to the topic name that you select in Step 6. If you want to change the topic name, you can move the pointer over the table and click Edit. You must specify a topic that exists in the Kafka cluster. For more information, see Rename an object to be synchronized.
Data format delivered to Kafka	The data that is synchronized to the Kafka cluster is stored in the Avro or SharePlex JSON format. For more information, see Data formats of a Kafka cluster.
Policy for Shipping Data to Kafka Partitions	Select a synchronization policy based on your business requirements. For more information, see Specify the policy for synchronizing data to Kafka partitions.
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.
	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
Retry Time for Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

- 9. In the lower-right corner of the page, click Next.
- 10. Select the initial synchronization type and filter options.

1.Select Source and Destination Instances for $>$		3.Advanced Settings			
Initial Synchronization: 🗹 Initial Incremental Data Synchronization Filter options: 🗹 Ignore DDL in incremental synchronization phase					
			Cancel	Previous Save	Precheck
Parameter	Description				

Parameter	Description			
Initialize synchronization	<b>Initial Incremental Data Synchronization</b> is selected by default. DTS synchronizes incremental data that is generated in the source database to the destination database.			
	<b>Ignore DDL in incremental synchronization phase</b> is selected by default. DTS does not synchronize DDL operations that are performed on the source database during incremental data synchronization.			
Filter options	<b>Note</b> The setting of this parameter does not take effect. DTS does not synchronize DDL operations that are performed on the source database regardless of whether you select this option.			

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1	> >>

# 10.Synchronize data to or from a DRDS instance

## 10.1. Synchronize data between PolarDB-X instances

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

#### Prerequisites

- A database is created in the source PolarDB-X instance based on one or more ApsaraDB RDS for MySQL instances.
- The tables to be synchronized contain primary keys.
- The destination database in the RDS instance that corresponds to the destination PolarDB-X instance has sufficient storage space.

#### Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- When the data synchronization task is running, do not upgrade or downgrade the PolarDB-X instance, migrate frequently-accessed tables, change shard keys, or perform DDL operations on source objects. Otherwise, the data synchronization task fails.
- If you switch the network type of the PolarDB-X instance during data synchronization, you must submit a ticket to update the network connection settings of the data synchronization task.

#### SQL operations that can be synchronized

INSERT, UPDATE, and DELETE

#### Before you begin

DTS does not support **schema synchronization** between PolarDB-X instances. Therefore, you must create a database and tables in the destination PolarDB-X instance based on the schemas of the objects in the source PolarDB-X instance. For more information, see Create a database and Create a table.

**Note** During schema synchronization, DTS synchronizes the schemas of the required objects from the source database to the destination database.

#### Procedure

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

**?** Note On the buy page, set both Source Instance and Destination Instance to PolarDB-X 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.

1.Select Source and De	stination 🔰 2.Select Ob				
Synchronization	Task Name: DRDS_TO_DRDS				
Source Instance Details					
Ins	stance Type: DRDS Instance				
Insta	nce Region: Singapore				
* DRDS	Instance ID: drds		•		
Destination Instance Details					
Ins	stance Type: DRDS Instance				
Insta	nce Region: Singapore				
* DRDS :	Instance ID: drds		•		
				Cancel Set Whitelist and Nex	xt
Section	Parameter	Description			
N/A	Synchronizatio n Task Name	DTS automatically you specify an info need to use a uniqu	generates a task nan rmative name to ider ue task name.	ne. We recommend that ntify the task. You do not	
	Instance Type	This parameter is s	et to DRDS Instance	e and cannot be changed.	
Source	Instance Region	The source region t change the value o	that you selected on f this parameter.	the buy page. You cannot	
Instance Details	DRDS Instance ID	Select the ID of the	e source PolarDB-X ins	stance.	

Section	Parameter	Description
	Instance Type	This parameter is set to <b>DRDS</b> Instance and cannot be changed.
Destination Instance	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.
Details	DRDS Instance ID	Select the ID of the destination PolarDB-X instance.

7.

8. Select the synchronization policy and the objects to be synchronized.

1.Select Source and Destination	n 2.Select Object to Be Synchronized		3.Advanced Settings	>	4.Precheck
Synchronization Mode	:One-Way Synchronization				
Available			Selected (To edit an object name e Edit.) Learn more.	or its filter, hover ov	er the object and click Hover over the object and click
Expand the tree before to get dtstest to get hhhtest to get test_1	you perform a glol 🛛	> <	tstestdata	٩	the dialog box appears, modi object name o destination da select the colu- migrate. To ensure com we will conver database nam lowercase by / you need to k uppercase, ple them manually
Select All Rename Databases and Table Select All	ables: <ul> <li>Do Not Change Database and Tal</li> </ul>	ble Names	Select All Change Database and Table Na	imes	
<ul> <li>Retry Time for Failed Con</li> </ul>	nection 720 Minutes 🥎				E
				Cancel	Previous Next
Setting	Description				

Setting	Description
Select the processing mode of conflicting tables	<ul> <li>Pre-check and Intercept: checks whether the destination tables are empty. If the destination tables are empty, the precheck is passed. If the tables are not empty, an error is returned during the precheck and the data synchronization task cannot be started.</li> <li>Ignore: skips the check for empty destination tables.</li> <li>Marning If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks.</li> <li>If the source and destination databases have the same schema, DTS does not synchronize the data records that have the same primary keys as the data records in the destination database.</li> <li>If the source and destination databases have different schemas, only some columns are synchronized or the data synchronization task fails.</li> </ul>
Select the objects to be synchronized	<ul> <li>Select one or more tables from the Available section and click the &gt; icon to move the tables to the Selected section.</li> <li>Note <ul> <li>You can select only tables as the objects to be synchronized.</li> <li>By default, after an object is synchronized to the destination instance, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Rename an object to be synchronized.</li> </ul> </li> </ul>
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.
Retry Time for Failed Connections	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.  ONOTE When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

#### 9. Click Next.

10. Specify whether you want to perform initial full data synchronization.

1.Select Source and Destination Instances for		3.Advanced Settings	A.Precheck
Initial Synchronization: 🗹 Initial Full Da	ta Synchronization		
			Cancel Previous Save Precheck

**?** Note During initial full data synchronization, DTS synchronizes the historical data of the required objects from the source database to the destination database. If you do not select Initial Full Data Synchronization, DTS does not synchronize the historical data.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the *icon next* to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1	> >>

## 10.2. Synchronize data from a PolarDB-X 1.0 instance to an AnalyticDB for MySQL cluster

is a real-time online analytical processing (OLAP) service that is developed by Alibaba Cloud for online data analysis with high concurrency. AnalyticDB for MySQL can analyze petabytes of data from multiple dimensions at millisecond-level timing to provide you with data-driven insights into your business. This topic describes how to synchronize data from a PolarDB-X 1.0 instance to an cluster by using Data Transmission Service (DTS). After you synchronize data, you can use AnalyticDB for MySQL to build internal business intelligence (BI) systems, interactive query systems, and real-time report systems.

#### Prerequisites

- A database is created in the source PolarDB-X instance based on one or more ApsaraDB RDS for MySQL instances.
- An cluster is created. For more information, see Create an cluster.
- The destination 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 loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- When the data synchronization task is running, do not upgrade or downgrade the PolarDB-X instance, migrate frequently-accessed tables, change shard keys, or perform DDL operations on source objects. Otherwise, the data synchronization task fails.
- If you switch the network type of the PolarDB-X instance during data synchronization, you must submit a ticket to update the network connection settings of the data synchronization task.
- We recommend that you do not use gh-ost or pt-online-schema-change to perform data definition language (DDL) operations on the required objects during data synchronization. Otherwise, data may fail to be synchronized.
- Due to the limits of , if the disk space usage of the nodes in an cluster reaches 80%, the cluster is locked. We recommend that you estimate the required disk space based on the objects that you want to synchronize. You must ensure that the destination cluster has sufficient storage space.
- Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to be synchronized.

### SQL operations that can be synchronized INSERT, UPDATE, and DELETE

#### Permissions required for database accounts

Database	Required permission
PolarDB-X	The SELECT permission on the objects to synchronize and the REPLICATION CLIENT and REPLICATION SLAVE permissions. These permissions are automatically granted by DTS.

Database	Required permission
	Read and write permissions on the objects to synchronize

#### Data type mappings

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

#### Procedure

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

On the buy page, set Source Instance to PolarDB-X, 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.

1.Select Source and Des	stination	> 2.Authori	ze AnalyticDB Account	> 3	.Select Obj	ject to Be Synchronized	$\geq$	4.Precheck
Synchronization	Task Name:	DRDS_To_ADB for M	ySQL					
Source Instance Details								
Inst	ance Type:	DRDS Instance						
Instar	nce Region:	China (Hangzhou)						
* DRDS I	nstance ID:	dra			•			
Destination Instance Details								
Inst	tance Type:	AnalyticDB	AnalyticDB					
Instar	nce Region:	China (Hangzhou)	China (Hangzhou)					
	*Version: 🔾 2.0 🔘 3.0							
	* Database: am-							
* Databas	se Account:	dtstest						
* Database	Password:	•••••			<b>4</b> >			
							Cancel	Set Whitelist and Next
Section	Section Parameter		Description					
N/A Synchronizatio n Task Name		The task nam that you spec the task. You	e that ify a d do not	DTS au escript t need	itomatically gen ive name that n to use a unique	nerates. W nakes it ea task nam	e recommend isy to identify e.	

Section	Parameter	Description
Source Instance Details	Instance Type	This parameter is set to <b>DRDS Instance</b> and cannot be changed.
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.
	DRDS Instance ID	The ID of the source PolarDB-X 1.0 instance.
	Instance Type	This parameter is set to <b>AnalyticDB</b> and cannot be changed.
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.
Destination	Version	Select 3.0.
Destination Instance Details	Database	The ID of the instance.
	Database Account	The database account of the cluster. For information about the permissions that are required for the account, see Permissions required for database accounts.
	Dat abase Password	The password of the database account.

#### 7.

8. Select the synchronization policy and objects to synchronize.

#### Data Synchronization Synchronize d

at a to or from a DRDS instance

1.Select Source and Destination Insta	nces 2.Authorize AnalyticDB Account	3.Select Object to Be Synchronized	4.Precheck
Initial Synchron	nization: 🗹 Initial Schema Synchronization 🔽 Initial F	ull Data Synchronization	
Note: do not clea cleans up the log	n up the incremental data log generated by the source datal too early, the DTS incremental task may fail	base after the DTS task is started when the DTS full task is running. If the source	Jatabase
Proccessing Existed Targe	Mode In t Table: Pre-check and Intercept $\bigcirc$ Ignore		
Merge Multi Synchronizatio	Tables:○ Yes ● No in Type:✔ Insert ✔ Update ✔ Delete		
Available		Selected (To edit an object name or its filter, hover over the obj Edit.) Learn more.	ect and click
Expand the tree	before you perform a glol 🔰 🍳	Q dtstestdata (20bjects)	
E Select All		> <	
*Rename Databases	and Tables: <ul> <li>O Not Change Database and T</li> </ul>	able Names O Change Database and Table Names	
* Retry Time for Fai	ed Connection 720 Minutes 🧭		
		Cancel Previ	ous Next Precheck
Setting	Description		
Initial Synchronization	You must select both <b>Init</b> <b>Synchronization</b> in most synchronizes the schemas to the destination cluster. incremental synchronization	ial Schema Synchronization and In cases. After the precheck is complete and data of required objects from the The schemas and data are the basis f on.	tial Full Data , DTS source instance or subsequent

Setting	Description				
Processing Mode In Existed Target Table	<ul> <li>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 those in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started.</li> <li>Note If the source and destination databases contain identical table names and the tables in the destination database cannot be deleted or renamed, you can use the object name mapping feature to rename the tables that are synchronized to the destination database. For more information, see Rename an object to be synchronized.</li> <li>Ignore Errors and Proceed: skips the precheck for identical table names in the source and destination databases.</li> <li>Marning If you select Ignore Errors and Proceed, data consistency is not guaranteed and your business may be exposed to potential risks.</li> <li>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.</li> <li>If the source and destination databases have different schemas, initial data synchronized, or the data synchronization task fails.</li> </ul>				
Merge Multi Tables	<ul> <li>If you select Yes, DTS adds thedts_data_source column to each table to record data sources.</li> <li>No is selected by default.</li> <li>Note If you set this parameter to Yes, all of the selected source tables in the task are merged into a destination table. If you want to merge only part of the source tables, you can create two data synchronization tasks.</li> </ul>				
Synchronization	Select the types of operations that you want to synchronize based on your business requirements. All operation types are selected by default. • Insert				
Туре	<ul><li> Update</li><li> Delete</li></ul>				
Setting	Description				
--------------------------------------	--	--	--	--	--
	Select one or more tables from the <b>Available</b> section and click the > icon to add the tables to the <b>Selected</b> section.				
Select the objects to synchronize	<ul> <li>Note</li> <li>You can select only tables as objects to synchronize.</li> <li>You can use the object name mapping feature to rename the columns that are synchronized to the destination database. For more information, see Rename an object to be synchronized.</li> </ul>				
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.				
Retry Time for	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.				
Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.				

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

1.Configure Source and Destination Instances 2			AnalyticDB Account 3.Select Objects to Synchronize			
AnalyticDB Table Group	AnalyticDB Table Name	Type(All) 👻	Primary Key Column	Distribution Column	Definition Status(All) 👻	
dtstestdata	customer	Partitioned 1 🔻	id	id 💌	Defined	
dtstestdata	order	Partitioned 1 🔻	orderid	orderid 🔻	Defined	
Set All to Partitioned Table	Set All to Dimension Table Enter a table name	search		Total: 2 item(s), Per Page: 2	$(0 \bullet)$ item(s) $(1 \circ)$ $(1 \circ)$	
				Cancel	Previous Save Precheck	

Note After you select Initial Schema Synchronization, you must specify the type, primary key column, and partition key column for the tables that you want to synchronize to . 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, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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



# 10.3. Synchronize data from a PolarDB-X 1.0 instance to an AnalyticDB for PostgreSQL instance

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

### Prerequisites

- A database is created in the source PolarDB-X instance based on one or more ApsaraDB RDS for MySQL instances.
- The tables to synchronize from the source database contain primary keys.
- An AnalyticDB for PostgreSQL instance is created. For more information, see Create a AnalyticDB for PostgreSQL instance.

### Precautions

• DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.

- When the data synchronization task is running, do not upgrade or downgrade the PolarDB-X instance, migrate frequently-accessed tables, change shard keys, or perform DDL operations on source objects. Otherwise, the data synchronization task fails.
- If you switch the network type of the PolarDB-X instance during data synchronization, you must submit a ticket to update the network connection settings of the data synchronization task.
- After a data synchronization task is started, new tables that are created in the source database can be synchronized. To do this, you must add the tables to the selected objects of the task. For more information, see Add an object to a data synchronization task.

# Limits

- Only tables can be selected as the objects to synchronize.
- DTS does not synchronize the following types of data: JSON, GEOMETRY, CURVE, SURFACE, MULT IPOINT, MULT ILINEST RING, MULT IPOLYGON, and GEOMET RYCOLLECT ION.

# SQL operations that can be synchronized

INSERT, UPDATE, and DELETE

# Term mappings

PolarDB-X	AnalyticDB for PostgreSQL
Database	Schema
Table	Table

## Procedure

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

On the buy page, set Source Instance to PolarDB-X, 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.

#### Data Synchronization Synchronize d at a to or from a DRDS instance

#### Dat a Transmission Service

1.Select Source and Destination T	nstances for	2.Select Object to Be Synchroniz	ed 🔉	3.Prech	eck
Synchronization Task Name:	DRDS_TO_ADB for PG				
	0.00_10_20010.110				
Source Instance Details					
Instance Type:	DRDS Instance				
Instance Region:	China (Hangzhou)				
* DRDS Instance ID:	drd:		•		
Destination Instance Details					
Instance Type:	AnalyticDB for PostgreSQL				
Instance Region:	China (Hangzhou)				
* Instance ID:	gp-				
* Database Name:	dtstestdata		]		
* Database Account:	dtstest		]		
* Database Password:	•••••	4			
			_		
				Cancel	Set Whitelist and Next

Section	Parameter	Description		
N/A	Synchronizatio n Task Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.		
	Instance Type	This parameter is set to <b>DRDS Instance</b> and cannot be changed.		
Source Instance Dotails	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.		
Details	DRDS Instance ID	The ID of the source PolarDB-X instance.		
	Instance Type	This parameter is set to <b>AnalyticDB for PostgreSQL</b> and cannob be changed.		
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.		
Destination	Instance ID	The ID of the AnalyticDB for PostgreSQL instance.		
Instance Details	Dat abase Name	The name of the destination database in the AnalyticDB for PostgreSQL instance.		
	Dat abase Account	The database account of the AnalyticDB for PostgreSQL instance.		
	Dat abase Password	The password of the database account.		

### 7.

8. Set the synchronization policy and objects to synchronize.

Initial Syr	chronization: 🔽 Initial Schema Synchronization	Initial Full Data Syr	chronization
Proccessing Mo Existed Target	ide In Table: O Clear Target Table Ignore		
Synchror	ization Type: 🗹 Insert 🕑 Update 🕑	Delete	
Available			Selected (To edit an object name or its filter, hover over the object and click
Expand the	tree before you perform a glol		Edit.) Learn more.
🗆 📑 dtstes	stdata		Q dtstestdata (10biects)
	order		i customer
		>	
		<	
Select All			
*Change Mapp	ed Name: 💿 Do Not Change D	atabase and Table Names	Select All <ul> <li>Change Database and Table Names</li> </ul>
*Change Mapp	ed Name:	atabase and Table Names	Select All  Change Database and Table Names  Cancel Previous
*Change Mapp	ed Name:	atabase and Table Names	Select All Cancel Previous
*Change Mapp	ed Name:	atabase and Table Names Description The synch both Initi Data Synch objects fr instance. subseque	Select All Cancel Previous Can

Set the Setting synchronizat	Parameter	Description			
ion policy	Processing Mode In Existed Target Table	<ul> <li>Clear Target Table         Skips the check for empty destination tables during the precheck. Clears the data in destination tables before initial full data synchronization. If you want to synchronize your business data after you test the data synchronization task, you can select this mode.     <li>Ignore         Skips the check for empty destination tables during the precheck. Synchronizes data to the destination tables without clearing the tables during initial full data synchronization. If you want to synchronize data from multiple tables to a single table, you can select this mode.     </li> </li></ul>			
	Synchronization Type	<ul> <li>The operations that you want to synchronize based on your business requirements.</li> <li>Insert</li> <li>Update</li> <li>Delete</li> </ul>			
		Select one or more tables from the <b>Available</b> section and click the icon to add the tables to the <b>Selected</b> section.			
Select the objects to synchronize	None	<ul> <li>Note</li> <li>You can select only tables as the objects to synchronize.</li> <li>You can use the object name mapping feature to rename the columns that are synchronized to the destination database. For more information, see Rename an object to be synchronized.</li> </ul>			
Rename Databases and Tables	None	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.			

Setting	Parameter	Description		
Retry Time	None	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.		
for Failed Connections		<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.		

- 9. In the lower-right corner of the page, click **Next**.
- 10. Specify the primary key columns and distribution columns of the tables that you want to synchronize to the AnalyticDB for PostgreSQL instance.

1.Configure	Source and Destination In	stances in 💦 🔪	2.Select Objects to Synchronize	3.Precheck
Schema	Table	Primary Key Column	Distribution Column	Definition Status(All) 👻
dtstestdata	customer	id	id 💌	Defined
dtstestdata	order	orderid	orderid •	Defined
dts.migration.mess	sage.greenplu Search			Total: 2 item(s), Per Page: 20 $\bullet$ item(s) « < 1 > »
				Cancel Previous Save Precheck

Onte The page in this step appears only if you select Initial Schema Synchronization in Step 8. For more information about primary key columns and distribution columns, see Define constraints and Define table distribution.

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

## ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the 🕧 icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscription Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1 > »

# 11.Synchronize data between Redis databases

# 11.1. Configure one-way data synchronization between ApsaraDB for Redis instances

Data Transmission Service (DTS) supports one-way data synchronization between ApsaraDB for Redis instances. This feature is suitable for 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.

**Warning** After you configure a data synchronization task, do not change the architecture type of the source or destination database. For example, if you change the master-replica architecture to the cluster architecture, data synchronization fails. For more information about the architecture types, see .

# Prerequisites

• The engine versions of the source and destination instances are supported by DTS. For more information, see Overview of data synchronization scenarios.

**?** Note The version of the destination instance must be the same as or later than that of the source instance. If you want to synchronize data between different versions of instances, make sure that the versions of the source and destination instances are compatible. For example, you can create a destination pay-as-you-go instance to verify the compatibility between the source and destination instances. Then, you can release this instance or change the billing method of the instance 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.
- If the source instance is a persistent memory-optimized instance of the ApsaraDB for Redis Enhanced Edition (Tair), the appendonly parameter must be set to ON.
- The source instance cannot be a storage-optimized instance of the ApsaraDB for Redis Enhanced Edition (Tair). In this scenario, a storage-optimized instance of ApsaraDB for Redis Enhanced Edition (Tair) can be used only as the destination instance.

# Precautions

- DTS uses the resources of the source and destination instances during initial full data synchronization. This may increase the loads of the database servers. If you migrate a large amount of data or if the server specifications cannot meet your requirements, database services may become unavailable. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination instances. We recommend that you synchronize data during off-peak hours.
- If an expiration policy is enabled for some keys in the source database, these keys may not be

deleted in a timely manner after they expired. Therefore, the number of keys in the destination database may be less than that in the source database. You can run the info command to view the number of keys in the destination database.

Onte The number of keys that do not have an expiration policy or have not expired is the same in the source and destination databases.

- If the engine version of the source instance is 2.8, incremental data synchronization is not supported.
- We recommend that you do not run the FLUSHDB or FLUSHALL command on the source instance during data synchronization. If you run one of the two commands, data inconsistency may occur between the source and destination instances.
- If the data eviction policy ( maxmemory-policy ) of the destination instance is not set to noeviction
   n , data inconsistency may occur between the source and destination instances. For more information about data eviction policies, see How does ApsaraDB for Redis evict data by default?
- During data synchronization, if the number of shards in the source instance is changed or if the database specifications are changed (for example, the memory capacity is scaled up), you must reconfigure the data synchronization task. To ensure data consistency, we recommend that you clear the data that has been synchronized to the destination instance before you reconfigure the task.
- During data synchronization, if the endpoint of the source instance is changed, you must submit a ticket to update the endpoint change. Instance operations that may cause endpoint changes include zone changes and network type changes from classic network to virtual private cloud (VPC). Otherwise, the append-only files (AOFs) of the source instance may be reset. For this reason, you must reconfigure the task.
- Limits on synchronizing a standalone ApsaraDB for Redis instance to an ApsaraDB for Redis cluster instance: Each command can be run only on a single slot in an ApsaraDB for Redis cluster instance. If you perform operations on multiple keys in the source database and the keys belong to different slots, an error occurs.

CROSSSLOT Keys in request don't hash to the same slot

We recommend that you perform operations on only one key during data synchronization. Otherwise, the synchronization task will be interrupted.

# Supported synchronization topologies

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

For more information, see Synchronization topologies.

# SQL operations that can be synchronized

- APPEND
- BIT OP, BLPOP, BRPOP, and BRPOPLPUSH
- DECR, DECRBY, and DEL
- EVAL, EVALSHA, EXEC, EXPIRE, and EXPIREAT
- GEOADD and GET SET
- HDEL, HINCRBY, HINCRBYFLOAT, HMSET, HSET, and HSETNX
- INCR, INCRBY, and INCRBYFLOAT

- LINSERT , LPOP, LPUSH, LPUSHX, LREM, LSET , and LT RIM
- MOVE, MSET, MSET NX, and MULT I
- PERSIST, PEXPIRE, PEXPIREAT, PFADD, PFMERGE, and PSETEX
- RENAME, RENAMENX, RESTORE, RPOP, RPOPLPUSH, RPUSH, and RPUSHX
- SADD, SDIFFST ORE, SELECT, SET, SET BIT, SET EX, SET NX, SET RANGE, SINTERST ORE, SMOVE, SPOP, SREM, and SUNIONST ORE
- ZADD, ZINCRBY, ZINTERSTORE, ZREM, ZREMRANGEBYLEX, ZUNIONSTORE, ZREMRANGEBYRANK, and ZREMRANGEBYSCORE
- SWAPDB and UNLINK (supported only if the engine version of the source instance is 4.0)

### ⑦ Note

- PUBLISH operations cannot be synchronized.
- If you run the EVAL or EVALSHA command to call Lua scripts, DTS cannot identify whether these Lua scripts are executed on the destination instance. This is because the destination instance does not explicitly return the execution results of Lua scripts during incremental data synchronization.
- When DTS runs the SYNC or PSYNC command to transfer data of the LIST type, DTS does not clear the existing data in the destination instance. As a result, the destination instance may contain duplicate data records.

# Permissions required for database accounts

Database	Permission and authorization method
Source ApsaraDB for Redis instance	The database account must have read permissions. For more information about how to grant the required permissions to an account, see Create and manage database accounts.
Destination ApsaraDB for Redis instance	The database account must have read and write permissions. For more information about how to grant the required permissions to an account, see Create and manage database accounts.

# Procedure

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

**Note** On the buy page, set both Source Instance and Destination 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 Task** in the Actions column.
- 6. Configure the source and destination instances.

#### Data Synchronization Synchronize d ata between Redis databases

#### Dat a Transmission Service

1.Configure Source and Destinatio	n 🔰 2.Select Objects to Synchronize 💡			
Synchronization Task Name:	Redis			
Source Instance Details				
Instance Type:	Redic Instance			
instance ryper	Kedis Instance			
Instance Region:	Singapore			
* Instance ID:	r-gs!	-		
Database Password:	***********	<⊅		
Destination Instance Details				
Instance Type:	Redis Instance	*		
Testerer Desirer	Circum and And Circum			
Instance Region:	Singapore			
* Instance ID:	r oct	_		
	1-93.	•		
Database Password:		(ه)		
		<b>T</b>		
			Cancel	Set Whitelist and Next
			Calicer	Det Triftenst and Next

Section	Parameter	Description
N/A	Synchronizatio n Task Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.
	Instance Type	Select Redis Instance.
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.
	Instance ID	The ID of the source instance.
Source Instance Details	Database Password	The database password of the source instance. For information about the permissions that are required for the database account, see Permissions required for database accounts. <b>Note</b> The database password is in the <user>: <password> format. For example, if the username of a custom account is admin and the password is Rp829dlwa, the database password is admin:Rp829dlwa.</password></user>
	Instance Type	Select Redis Instance.
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.
	Instance ID	The ID of the destination instance.
Destination		

Pestimation		
betabe Details	Parameter	Description
		The database password of the destination instance. For information about the permissions that are required for the database account, see Permissions required for database accounts.
	Database Password	<b>Note</b> The database password is in the <user>: <password> format. For example, if the username of a custom account is admin and the password is Rp829dlwa, the database password is admin:Rp829dlwa.</password></user>

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

#### ? Note

- You do not need to modify the security settings for ApsaraDB instances (such as and ) and databases that are hosted on Elastic Compute Service (ECS). DTS automatically adds the CIDR blocks of DTS servers to the whitelists of ApsaraDB instances or the security group rules of ECS instances. For more information, see Add the CIDR blocks of DTS servers to the security settings of on-premises databases.
- After data synchronization is complete, we recommend that you remove the CIDR blocks of DTS servers from the whitelists or security groups.
- 8. Select the processing mode of conflicting tables and the objects to synchronize.

	2. Select Objects to Synchronize		3.Advanced Settings		4.Precheck
Synchronization Mode:One-Way S	ynchronization				
Proccessing Mode In Existed Target Table:	e-check and Intercept $ \odot $ Ignore				
Available	expand the L		Selected (To edit an object name or Edit.) Learn more.	its filter, hover over the	object and click
► 1 ► 2			0	Q	
≌ 3 ≌ 4 ≌ 5		>			
<b>6</b> <b>7</b>		<			
► 11 ► 12 ► 13					
i 13 i 14 Select All	•				
*Name batch change:	No Yes		Select All		
					Canada

Setting	Description
Select the processing mode of conflicting tables	<ul> <li>Pre-check and Intercept: checks whether the destination instance is empty. If the destination instance is empty, the precheck is passed. If the instance is not empty, an error is returned during the precheck and the data synchronization task cannot be started.</li> <li>Ignore: skips the check for empty destination instances.</li> <li>Warning If you select Ignore, data records in the source instance overwrite the data records that have the same keys in the destination instance. Proceed with caution.</li> </ul>
Select the objects to synchronize	<ul> <li>Select one or more databases from the Available section and click the icon to add the databases to the Selected section.</li> <li>You can select only databases as objects to synchronize. Keys cannot be selected as objects to synchronize.</li> </ul>
Rename Databases and Tables	In this scenario, you cannot rename objects.
Replicate Temporary Tables When DMS Performs DDL Operations	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> <li>Note If online DDL operations generate a large amount of data, the data synchronization task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> <li>Note If you select No, the tables in the destination database may be locked.</li> </ul>

Setting	Description					
	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.					
Retry Time for Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.					

- 9. In the lower-right corner of the page, click **Next**.
- 10. Select initial synchronization types. The value is set to **Include full data + increment al data** and cannot be changed.

1.Configure Source and Destination $ig>$	2.Select Objects to Synchronize	3.Advanced Settings		4.Precheck
Initial Synchronization: <b>Includ</b>	e full data + incremental data			
		Cancel	Previous	Save Precheck

### ? Note

- DTS synchronizes historical data from the source instance to the destination instance. Then, DTS synchronizes incremental data.
- If a version-related error message is displayed, you must upgrade the source instance to a specified version. For more information about how to upgrade the version, see Upgrade the major version and Update the minor version.

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

### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- You can troubleshoot the issues based on the causes and run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1	> >>

Onte You can view the state of the data synchronization task on the Synchronization Tasks page.

# 11.2. Configure two-way data synchronization between ApsaraDB for Redis Enhanced Edition (Tair) instances

This topic describes how to configure two-way data synchronization between ApsaraDB for Redis Enhanced Edition (Tair) instances by using Data Transmission Service (DTS). The data synchronization feature is applicable to scenarios such as active geo-redundancy and geo-disaster recovery.

# Prerequisites

The source and destination instances are ApsaraDB for Redis Enhanced Edition (Tair) instances (version 5.0).

## ? Note

- If the ApsaraDB for Redis instance is an Enhanced Edition instance (storage-optimized instance), it can be used only as the destination database.
- If the source ApsaraDB for Redis instance is an Enhanced Edition instance (persistent memory-optimized instance), you must set the appendonly parameter to yes.
- ApsaraDB for Redis Enhanced Edition (Tair) supports the **cluster**, **standard**, and **read/write splitting** architectures.

## Precautions

• During two-way data synchronization, the data synchronization task in the forward direction performs and . The data synchronization task in the reverse direction performs only incremental data synchronization.

**Warning** To ensure data consistency, do not modify or write data to the same key in the source and destination databases when the two-way data synchronization tasks are running.

#### full data synchronizationincremental data synchronization

- DTS uses the resources of the source and destination databases during full data synchronization. This may increase the loads of the database servers. If you synchronize a large volume of data or the server specifications cannot meet your requirements, the database services may become unavailable. Before you synchronize data, evaluate the impacts of data synchronization on the performance of the source and destination instances. We recommend that you synchronize data during off-peak hours.
- We recommend that you do not run the FLUSHDB or FLUSHALL command in the source instance during data synchronization. Otherwise, data may become inconsistent between the source and destination instances.
- •
- If an expiration policy is enabled for some keys in the source database, these keys may not be deleted in a timely manner after they expired. Therefore, the number of keys in the destination database may be less than that in the source database. You can run the info command to view the number of keys in the destination database.

**?** Note The number of keys that do not have an expiration policy or have not expired is the same in the source and destination databases.

• If direct connection is disabled for the destination ApsaraDB for Redis instance, DTS uses the proxy forwarding mode to write data to the destination instance.

Onte For more information about how to enable direct connection, see Enable the direct connection mode.

- During data synchronization, if the number of shards in the source or destination ApsaraDB for Redis instance is increased or decreased, or if the database specifications are changed (for example, the memory capacity is scaled up), you must reconfigure the task. To ensure data consistency, we recommend that you clear the data that has been synchronized to the source and destination Redis databases before you reconfigure the task.
- During data synchronization, if the endpoint of the source or destination ApsaraDB for Redis instance is changed (for example, the zone of the instance is changed or the network type is changed from classic network to VPC), you must submit a ticket to update the change. Otherwise, the append-only files (AOF) of the source or destination ApsaraDB for Redis instance may be reset. In this case, you must reconfigure the task.
- If the source or destination instance of a two-way data synchronization task resides in a region outside the Chinese mainland, data can be synchronized only within this region. Cross-region two-way synchronization is not supported. For example, if the source instance resides in the Japan (Tokyo) region, data can be synchronized only within the Japan (Tokyo) region and cannot be synchronized to the Germany (Frankfurt) region.
- Limits on synchronizing a standalone ApsaraDB for Redis instance to an ApsaraDB for Redis cluster instance: Each command can be run only on a single slot in an ApsaraDB for Redis cluster instance. If you perform operations on multiple keys in the source database and the keys belong to different slots, an error occurs.

CROSSSLOT Keys in request don't hash to the same slot

We recommend that you perform operations on only one key during data synchronization. Otherwise, the synchronization task will be interrupted.

# Operations that can be synchronized

- APPEND
- BIT OP, BLPOP, BRPOP, and BRPOPLPUSH
- DECR, DECRBY, and DEL
- EVAL, EVALSHA, EXEC, EXPIRE, and EXPIREAT
- GEOADD and GET SET
- HDEL, HINCRBY, HINCRBYFLOAT, HMSET, HSET, and HSETNX
- INCR, INCRBY, and INCRBYFLOAT
- LINSERT, LPOP, LPUSH, LPUSHX, LREM, LSET, and LT RIM
- MOVE, MSET, MSET NX, and MULTI
- PERSIST, PEXPIRE, PEXPIREAT, PFADD, PFMERGE, and PSETEX
- RENAME, RENAMENX, RPOP, RPOPLPUSH, RPUSH, and RPUSHX
- SADD, SDIFFST ORE, SELECT, SET, SET BIT, SET EX, SET NX, SET RANGE, SINT ERST ORE, SMOVE, SPOP, SREM, and SUNIONST ORE
- UNLINK, ZADD, ZINCRBY, ZINTERSTORE, ZREM, ZREMRANGEBYLEX, ZUNIONSTORE, ZREMRANGEBYRANK, and ZREMRANGEBYSCORE
- SWAPDB (This operation is not supported if the source or destination ApsaraDB for Redis instance is deployed in the cluster architecture.)

#### ♥ Notice

- PUBLISH operations cannot be synchronized.
- If you run the EVAL or EVALSHA command to call Lua scripts, DTS cannot identify whether these Lua scripts are executed on the destination database. During incremental data synchronization, the destination database does not explicitly return the execution results of Lua scripts.
- When DTS runs the SYNC or PSYNC command to transfer data of the LIST type, DTS does not clear the existing data. As a result, the destination instance may contain duplicate data records.

Database	Permissions and authorization method		
Source ApsaraDB for Redis instance	The database accounts of the source and destination instances must have the read and write permissions. For more information about how		
Destination ApsaraDB for Redis instance	to authorize a database account, see Create and manage database accounts.		

# Permissions required for database accounts

## Procedure

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

Notice On the buy page, set Source Instance to Redis, set Destination Instance to Redis, and set Synchronization Topology to Two-way Synchronization.

- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click **Data Synchronization**.
- 4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.
- 5. Configure the data synchronization task in the forward direction.
  - i. Find the data synchronization instance, and click **Configure Task** in the **Actions** column of the first data synchronization task.

Notice A two-way data synchronization instance contains two data synchronization tasks. You must set parameters for each task. When you configure the second data synchronization task, find the task and click **Configure Task** in the **Actions** column.

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
And the second			Pay-As-You-Go	Two-Way Synchronization	Switch to Subscription Upgrade View Synchronization Task~ More
Task Name	Status	Synchronization Details	Source/Destination Instance		Actions
-	Not Configured		Not Configured Not Configured	Co	nfigure Synchronization Channel
ingen ungen under	Not Configured		Not Configured Not Configured	Co	nfigure Synchronization Channel

#### ii. Configure the source and destination instances.

1.Configure Source and De	stination 2.Select C	Objects to Synchronize	>	3.Advanced Settings	>	4.Precheck
Synchronization Task	Name: Redis_Forward					
Source Instance Details						
Instance	Type: Redis Instance		Ŧ			
Instance R	egion: China (Beijing)					
* Instan	ice ID: r-;		-			
Database Pass	sword:		<⊅			
Destination Instance Details						
Instance	Type: Redis Instance		v			
Instance R	egion: China (Beijing)					
* Instan	ice ID: r-		-			
Database Pass	sword:		<4>			
						Cancel Set Whitelist and Next
Soction	Daramotor	Description				
Section	Faldilletei	Description				
N/A	Synchronizati on Task Name	DTS automation you specify an not need to us	ally g inforr se a ur	enerates a task n native name to ic nique task name.	ame. W lentify t	e recommend that he task. You do

Section	Parameter	Description
	Instance Type	Select Redis Instance.
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.
Source Instance Details		Select the ID of the source ApsaraDB for Redis instance.
	Instance ID	<b>Notice</b> When you configure the data synchronization task in the reverse direction, select the ID of the destination ApsaraDB for Redis instance.
	Dat abase Password	Enter the database password of the ApsaraDB for Redis instance. For information about the permissions that are required for the account, see Permissions required for database accounts.
		Notice
	Instance Type	Select Redis Instance.
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.
		Select the ID of the destination ApsaraDB for Redis instance.
Destination Instance Details	Instance ID	• Notice When you configure the data synchronization task in the reverse direction, select the ID of the source ApsaraDB for Redis instance.
		Enter the database password of the ApsaraDB for Redis
	Database	instance. For information about the permissions that are required for the account, see Permissions required for database accounts.
		♥ Notice

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

#### ? Note

- You do not need to modify the security settings for ApsaraDB instances (such as and ) and ECS-hosted databases. DTS automatically adds the CIDR blocks of DTS servers to the whitelists of ApsaraDB instances or the security group rules of Elastic Compute Service (ECS) instances. For more information, see Add the CIDR blocks of DTS servers to the security settings of on-premises databases.
- After data synchronization is completed, we recommend that you remove the CIDR blocks of DTS servers from the whitelists or security groups.
- iv. Select the synchronization policy and the objects to be synchronized.

Configure Source a	nd Destination 2.Select Ob	jects to Synchronize		3.Advanced Settings	>	4.Precheck
Sync	hronization Mode:Two-Way Synchronization	1				
Conf	lict Resolution Policy: Overwrite (When a	conflict accurs the confl	icting second	in the V		
com	Overwrite (wrien a	connict occurs, the conn	icung record	in di ·		
Exist	ed Target Table:       Pre-check and Inter	cept 🔍 Ignore				
Ava	ilable			Selected		
If	you search globally, please expand   Q	<b>^</b>			Q	
-	1			<b>i</b> 0		
-	2					
	4					
9	5		>			
	7		<			
-	8					
-	9					
	11					
-	12					
	15	-				
Sel	ect All			Select All		
					Car	ncel Previous
Setting	Parameter	Descriptio	n			
		Overwritte overwritte	e (Conf en)	licting records in th	e destinatio	on instance are
	Conflict Resolution Policy	During dat but differ overwrites	a sync ent vali s the co	hronization, if data ues, the data record onflicting records.	records hav d with the la	ve the same key atest key value
`oloct						
select						

the Setting synchroni	Parameter	Description
zation policy	Select the processing mode of conflicting tables	<ul> <li>Pre-check and Intercept: checks whether the destination instance is empty. If the destination database is empty, the precheck is passed. If the database is not empty, an error is returned during the precheck and the data synchronization task cannot be started.</li> <li>Ignore: skips the check for empty destination databases.</li> <li>Marning If you select Ignore, the data records in the source database overwrite the data records with the same keys in the destination database. Proceed with caution.</li> </ul>
Select the objects to be synchroni zed	N/A	<ul> <li>Select one or more databases from the Available section and click icon to move the databases to the Selected section.</li> <li>You can select only databases as the objects to be synchronized. You cannot select keys as the objects to be synchronized.</li> </ul>
Rename Database s and Tables	N/A	In this scenario, you cannot rename objects.
Replicate Tempora ry Tables When DMS Performs DDL Operatio ns	N/A	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> <li>Yote If online DDL operations generate a large amount of data, the data synchronization task may be delayed.</li> <li>No: DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.</li> <li>Note If you select No, the tables in the destination database may be locked.</li> </ul>

Setting	Parameter	Description	
Retry Time for Failed Connecti ons	N/A	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.	
		<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.	

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

		onize	3.Advanced Settings	
Initial Synchronization: <b>Include f</b> the same process during forward synchronizat	ull data + incremental data ion.)	(Tables that have (	experienced initial synchronization during re	verse synchronization will not go through
			Cancel	revious Save Precheck

The value is set to **Include full data + incremental data**. DTS synchronizes historical data from the source ApsaraDB for Redis instance to the destination ApsaraDB for Redis instance. Then, DTS synchronizes incremental data.

### ♥ Notice

- If all the required objects have been synchronized from the source instance to the destination instance, the data synchronization task in the reverse direction synchronizes only incremental data.
- If a version-related error message appears, you can upgrade the source ApsaraDB for Redis instance to a specified version. For more information, see Upgrade the major version and Update the minor version.

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

? Note

- Before you can start the data synchronization task, DTS performs a precheck. You
  can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed

item to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.
- viii. Close the **Precheck** dialog box after the following message is displayed: **Precheck Passed**. Then, the data synchronization task in the forward direction starts.
- 6. Wait until initial synchronization is completed and the data synchronization task is in the **Synchronizing** state.

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

- 7. Configure the data synchronization task in the reverse direction.
  - i. Find the second data synchronization task, and click **Configure Synchronization Channel** in the Actions column.

8000-000			Pay-As-You-Go	Switch to Subscription   Two-Way Upgrade Synchronization View Synchronization Task ~   More
Task Name	Status	Synchronization Details	Source/Destination Instance	Actions
Redis_Forward	Completed		r- eo r- ndm	Pause Task   More
Redis_Reverse	Not Configured		r- ndm r- eo	Configure Synchronization Channel

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

## Result

After a period of time, both data synchronization tasks are in the **Synchronizing** state.

Syn	chronization Task ID 🔻		Search Sort:	Default Sorting	Status: All	
	Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) -	Actions
	dts			Pay-As-You-Go	Two-Way Synchronization	Switch to Subscription   Upgrade View Synchronization Task~   More
	Task Name	Status	Synchronization Details	Source/Destination Instan	ісе	Actions
	Redis_Forward	Synchronizing	Delay: 0 Milliseconds Speed: 0.00TPS/(0.000MB/s)	r- eo r- ndm		Pause Task More
	Redis_Reverse	Synchronizing	Delay: 0 Milliseconds Speed: 0.00TPS/(0.000MB/s)	r- ndm ) r- eo		Pause Task More

# 11.3. Use OpenAPI Explorer to configure one-way or two-way data synchronization between ApsaraDB for Redis Enhanced Edition (Tair) instances

You can call the API operations of Data Transmission Service (DTS) in OpenAPI Explorer to achieve oneway or two-way data synchronization between ApsaraDB for Redis Enhanced Edition (Tair) instances. This topic describes the precautions and provides sample code.

The sample code provided in this topic is written in Java. You must replace <accessKeyId> and <accessKeyId> in the sample code with your AccessKey ID and AccessKey secret. OpenAPI Explorer automatically calculates the signature value. For your convenience, we recommend that you call API operations in OpenAPI Explorer. After you call an operation, OpenAPI Explorer dynamically generates the sample code of the operation for SDKs for different programming languages.

# Prerequisites

The source and destination instances are ApsaraDB for Redis Enhanced Edition (Tair) instances (version 5.0).

## ? Note

- For one-way synchronization, the source instance cannot be a storage-optimized instance. For two-way synchronization, the source and destination instances cannot be storageoptimized instances.
- •
- ApsaraDB for Redis Enhanced Edition (Tair) supports the **cluster**, **standard**, and **read/write splitting** architectures.

# Usage notes

• During two-way data synchronization, the data synchronization task in the forward direction performs and . The data synchronization task in the reverse direction performs only incremental data synchronization.

**Warning** To ensure data consistency, do not modify or write data to the same key in the source and destination databases when the two-way data synchronization tasks are running.

### full data synchronizationincremental data synchronization

• DTS uses the resources of the source and destination databases during full data synchronization. This may increase the loads of the database servers. If you synchronize a large volume of data or the server specifications cannot meet your requirements, the database services may become unavailable. Before you synchronize data, evaluate the impact of data synchronization on the performance of the source and destination instances. We recommend that you synchronize data during off-peak hours.

- We recommend that you do not run the FLUSHDB or FLUSHALL command in the source instance during data synchronization. Otherwise, data may become inconsistent between the source and destination instances.
- •
- If an expiration policy is enabled for some keys in the source database, these keys may not be deleted in a timely manner after they expired. Therefore, the number of keys in the destination database may be less than that in the source database. You can run the info command to view the number of keys in the destination database.

**?** Note The number of keys that do not have an expiration policy or have not expired is the same in the source and destination databases.

• If direct connection is disabled for the destination ApsaraDB for Redis instance, DTS uses the proxy forwarding mode to write data to the destination instance.

Onte For more information about how to enable direct connection, see Enable the direct connection mode.

•

- •
- If the source or the destination instance is located in a region outside the Chinese mainland, two-way data synchronization is supported only between instances located with the same region. For example, two-way data synchronization is supported between instances within the Japan (Tokyo) region. Two-way data synchronization between an instance in the Japan (Tokyo) region and another instance in the Germany (Frankfurt) region is not supported.
- Limits on synchronizing a standalone ApsaraDB for Redis instance to an ApsaraDB for Redis cluster instance: Each command can be run only on a single slot in an ApsaraDB for Redis cluster instance. If you perform operations on multiple keys in the source database and the keys belong to different slots, an error occurs.

CROSSSLOT Keys in request don't hash to the same slot

We recommend that you perform operations on only one key during data synchronization. Otherwise, the synchronization task will be interrupted.

# Limits on features

One-way cascade synchronization is not supported. For more information, see Synchronization topologies.

# Operations that can be synchronized

- APPEND
- BIT OP, 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 LT RIM

- MOVE, MSET, MSET NX, and MULTI
- PERSIST, PEXPIRE, PEXPIREAT, PFADD, PFMERGE, and PSETEX
- RENAME, RENAMENX, RPOP, RPOPLPUSH, RPUSH, and RPUSHX
- SADD, SDIFFST ORE, SELECT, SET, SET BIT, SET EX, SET NX, SET RANGE, SINTERST ORE, SMOVE, SPOP, SREM, and SUNIONST ORE
- ZADD, ZINCRBY, ZINTERSTORE, ZREM, ZREMRANGEBYLEX, ZUNIONSTORE, ZREMRANGEBYRANK, and ZREMRANGEBYSCORE
- SWAPDB (This operation is not supported if the source or destination ApsaraDB for Redis instance is deployed in the cluster architecture.)

♥ Notice

- PUBLISH operations cannot be synchronized.
- If you run the EVAL or EVALSHA command to call Lua scripts, DTS cannot identify whether these Lua scripts are executed on the destination database. During incremental data synchronization, the destination database does not explicitly return the execution results of Lua scripts.
- When DTS runs the SYNC or PSYNC command to transfer data of the LIST type, DTS does not clear the existing data. In this case, the destination database may contain duplicate data records.

### Preparations

1. Create an AccessKey pair. For more information, see Create an AccessKey pair.

(?) Note To protect the AccessKey pair of your Alibaba Cloud account, we recommend that you create a RAM user, grant the RAM user the permissions to access DTS, and then use the AccessKey pair of the RAM user to call DTS SDK for Java. For more information, see Implement access control by using RAM.

2. Download SDKs. For more information, see Use DTS SDK for Java.

#### Create a data synchronization task

For more information about the request parameters and response parameters in this example, see CreateSynchronizationJob.

(?) Note If you set the Topology parameter to *oneway*, a one-way data synchronization task is created. If you set the parameter to *bidirectional*, two-way data synchronization tasks are created.

Sample request:

import com.aliyuncs.DefaultAcsClient; import com.aliyuncs.IAcsClient; import com.aliyuncs.exceptions.ClientException; import com.aliyuncs.exceptions.ServerException; import com.aliyuncs.profile.DefaultProfile; import com.google.gson.Gson; import java.util.\*; import com.aliyuncs.dts.model.v20180801.\*; public class CreateSynchronizationJob { public static void main(String[] args) { DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessKeyId>", "<accessSecret>"); IAcsClient client = new DefaultAcsClient(profile); CreateSynchronizationJobRequest request = new CreateSynchronizationJobRequest(); request.setRegionId("cn-hangzhou"); request.setSourceRegion("cn-beijing"); request.setDestRegion("cn-beijing"); request.setSynchronizationJobClass("small"); request.setPayType("Postpaid"); request.setTopology("oneway"); request.setSourceEndpointInstanceType("Redis"); request.setDestinationEndpointInstanceType("Redis"); try { CreateSynchronizationJobResponse response = client.getAcsResponse(request); System.out.println(new Gson().toJson(response)); } catch (ServerException e) { e.printStackTrace(); } catch (ClientException e) { System.out.println("ErrCode:" + e.getErrCode()); System.out.println("ErrMsg:" + e.getErrMsg()); System.out.println("RequestId:" + e.getRequestId()); } } }

#### Sample response:

```
{
    "SynchronizationJobId": "dts******",
    "RequestId": "158347****-rqyjQ",
    "Success": true
}
```

# Configure a one-way data synchronization task

In this example, the database account of the source instance must have the read permissions on the source database. The database account of the destination instance must have the read and write permissions on the destination database.

**Note** When you configure a one-way data synchronization task, you can also use a self-managed Redis database or an ApsaraDB for Redis Community Edition instance as the source or destination database. The configuration method is similar to that described in this topic. However, you must specify the request parameters based on your actual scenario, such as MigrationReserved.

# For more information about the request parameters and response parameters, see ConfigureSynchronizationJob.

#### Sample request:

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.dts.model.v20180801.*;
public class ConfigureSynchronizationJob {
   public static void main(String[] args) {
       DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessKeyId>",
"<accessSecret>");
       IAcsClient client = new DefaultAcsClient(profile);
        ConfigureSynchronizationJobRequest request = new ConfigureSynchronizationJobRequest
();
        request.setRegionId("cn-hangzhou");
        request.setSynchronizationJobId("dts*****");
        request.setStructureInitialization(true);
        request.setDataInitialization(true);
        request.setSynchronizationObjects("[{\"DBName\": \"0\",\"NewDBName\": \"0\"}]");
        request.setSynchronizationJobName("apitest");
        request.setSourceEndpointInstanceId("r-2ze******");
        request.setSourceEndpointInstanceType("redis");
        request.setSourceEndpointPassword("Test*****");
        request.setDestinationEndpointInstanceId("r-2ze******");
        request.setDestinationEndpointInstanceType("redis");
        request.setDestinationEndpointPassword("Test****");
        request.setMigrationReserved("{\"srcRedisType\":\"enterprise\",\"destRedisType\":\"
enterprise\"}");
        try {
            ConfigureSynchronizationJobResponse response = client.getAcsResponse(request);
           System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
           e.printStackTrace();
        } catch (ClientException e) {
           System.out.println("ErrCode:" + e.getErrCode());
           System.out.println("ErrMsg:" + e.getErrMsg());
           System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

#### Sample response:

```
{
    "RequestId": "1583*****-BcO5F",
    "Success": true
}
```

# Configure two-way data synchronization tasks

In this example, the database accounts used for data synchronization must have the read and write permissions.

For more information about the request parameters and response parameters, see ConfigureSynchronizationJob.

Sample request:

1. Configure the data synchronization task in the forward direction.

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.dts.model.v20180801.*;
public class ConfigureSynchronizationJob {
    public static void main(String[] args) {
        DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessKeyId
>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        ConfigureSynchronizationJobRequest request = new ConfigureSynchronizationJobReq
uest();
        request.setRegionId("cn-hangzhou");
        request.setSynchronizationJobId("dts******");
        request.setStructureInitialization(true);
        request.setDataInitialization(true);
        request.setSynchronizationObjects("[{\"DBName\": \"0\",\"NewDBName\": \"0\"}]")
;
        request.setSynchronizationJobName("apitest");
        request.setSynchronizationDirection("Forward");
        request.setSourceEndpointInstanceId("r-2ze******");
        request.setSourceEndpointInstanceType("redis");
        request.setSourceEndpointPassword("Test*****");
        request.setDestinationEndpointInstanceId("r-2ze******");
        request.setDestinationEndpointInstanceType("redis");
        request.setDestinationEndpointPassword("Test*****");
        request.setMigrationReserved("{\"srcRedisType\":\"enterprise\",\"destRedisType\
":\"enterprise\"}");
        try {
            ConfigureSynchronizationJobResponse response = client.getAcsResponse(reques
t);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

2. Wait until the data synchronization task in the forward direction enters the Synchronizing state.

(?) Note You can view the status of a data synchronization task in the DTS console. You can also call the DescribeSynchronizationJobStatus operation to view the status.

3. Configure the data synchronization task in the reverse direction.

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.dts.model.v20180801.*;
public class ConfigureSynchronizationJob {
   public static void main(String[] args) {
       DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessKeyId
>", "<accessSecret>");
       IAcsClient client = new DefaultAcsClient(profile);
       ConfigureSynchronizationJobRequest request = new ConfigureSynchronizationJobReq
uest();
        request.setRegionId("cn-hangzhou");
        request.setSynchronizationJobId("dts******");
        request.setStructureInitialization(true);
        request.setDataInitialization(true);
        request.setSynchronizationObjects("[{\"DBName\": \"0\",\"NewDBName\": \"0\"}]")
;
       request.setSynchronizationJobName("apitest");
       request.setSynchronizationDirection("Reverse");
        request.setSourceEndpointInstanceId("r-2ze******");
        request.setSourceEndpointInstanceType("redis");
        request.setSourceEndpointPassword("Test*****");
        request.setDestinationEndpointInstanceId("r-2ze******");
        request.setDestinationEndpointInstanceType("redis");
        request.setDestinationEndpointPassword("Test*****");
        request.setMigrationReserved("{\"srcRedisType\":\"enterprise\",\"destRedisType\
":\"enterprise\"}");
       try {
            ConfigureSynchronizationJobResponse response = client.getAcsResponse(reques
t);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
   }
}
```

#### Sample response:

```
{
    "RequestId": "1583******-Bc05F",
    "Success": true
}
```

## References

- View the connection status and performance of data synchronization You can view the performance of data synchronization by using the performance metrics. DTS provides the connection and performance metrics to help you manage data synchronization tasks.
- Add an object to a data synchronization task/Remove an object from a data synchronization task During a data synchronization task, you can add objects to or remove objects from the task.

# 11.4. Synchronize data from a selfmanaged Redis database hosted on ECS to an ApsaraDB for Redis instance

Data Transmission Service (DTS) supports one-way data synchronization between 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 self-managed Redis database hosted on Elastic Compute Service (ECS) to an ApsaraDB for Redis instance.



# Prerequisites

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

⑦ 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.
- If the source Redis database is deployed in a cluster architecture, all nodes of the Redis cluster must support the PSYNC command and share the same password.

# Precautions

- •
- If an expiration policy is enabled for some keys in the source database, these keys may not be deleted in a timely manner after they expired. Therefore, the number of keys in the destination database may be less than that in the source database. You can run the info command to view the number of keys in the destination database.

**?** Note The number of keys that do not have an expiration policy or have not expired is the same in the source and destination databases.

- If the bind parameter is configured in the *redis.conf* file of the source database, you must set the value of this parameter to the internal IP address of the ECS instance. The setting ensures that DTS can connect to the source database.
- To ensure the stability of data synchronization, we recommend that you increase the value of the *re pl-backlog-size* parameter in the redis.conf file of the source Redis database.
- To ensure the synchronization quality, DTS adds the following key to the source Redis database: DT S\_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 database during data synchronization. Otherwise, data may become inconsistent between the source and destination databases.
- •
- During data synchronization, if the number of shards in the self-managed Redis database is increased or decreased, or if the specifications of the database are changed (for example, the memory capacity is scaled up), you must reconfigure the task. To ensure data consistency, we recommend that you clear the data that has been synchronized to the destination Redis database before you reconfigure the task.
- During data synchronization, if the endpoint of the self-managed Redis database is changed, you must submit a ticket to update the change. Otherwise, the append-only files (AOF) of the self-managed Redis database may be reset. In this case, you must reconfigure the task.
- Limits on synchronizing a standalone ApsaraDB for Redis instance to an ApsaraDB for Redis cluster instance: Each command can be run only on a single slot in an ApsaraDB for Redis cluster instance. If you perform operations on multiple keys in the source database and the keys belong to different slots, an error occurs.

CROSSSLOT Keys in request don't hash to the same slot

We recommend that you perform operations on only one key during data synchronization. Otherwise, the synchronization task will be interrupted.

## Procedure

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

(?) Note On the buy page, set both Source Instance and Destination 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.

#### Data Synchronization Synchronize d

1.Configure Source and Destination	n 2.Select Objects to Synchronize	$\geq$	3.Advanced Settings	$\geq$	4.Precheck
Synchronization Task Name:	Redis				
Source Instance Details					
Source Instance Details					
Instance Type:	User-Created Database in ECS Instance	v			
Instance Region:	Singapore				
* ECS Instance ID:	and a second	-			
Database Type:	Redis				
Instance Mode:	Standalone      Cluster				
* Port Number:	7000				
Database Password:	•••••	<b>4</b> >			
Destination Instance Details					
Instance Type:	Redis Instance	•			
Instance Region:	Singapore				
* Instance ID:	r-gs5	•			
Database Password:	******	<b>\$</b> >			
					Cancel Set Whitelist and Next

Section	Parameter	Description		
N/A	Synchroniz at io n T ask Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.		
	Instance Type	Select User-Created Database in ECS Instance.		
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.		
	ECS Instance ID	Select the ID of the ECS instance that hosts the source Redis database.		
		<b>Note</b> 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 <b>Redis</b> .		
	Instance Mode	Select <b>Standalone</b> or <b>Cluster</b> based on the architecture of the source Redis database.		
Source Instance Details				

Section	Parameter	Description		
	Port Number	Enter the service port number of the source Redis database. The default port number is <b>6379</b> . In this example, enter <b>7000</b> .		
		<b>Note</b> 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 of the source Redis database.		
		<b>Note</b> This parameter is optional and can be left blank if no database password is set.		
	Instance Type	Select Redis Instance.		
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.		
Destination Instance Details	Instance ID	Select the ID of the destination Redis instance.		
	Dat abase Password	Enter the database password of the destination ApsaraDB for Redis instance.		
		⑦ Note		

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

**?** Note DTS adds the CIDR blocks of DTS servers to the inbound rule of the source ECS instance and the whitelist of the destination ApsaraDB for Redis instance. This ensures that DTS servers can connect to the source and destination instances.

8.

9.

10.

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

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- You can trouble shoot the issues based on the causes and run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.

# 12.

13.

# 11.5. Synchronize data from a selfmanaged Redis database connected over Express Connect, VPN Gateway, or Smart Access Gateway to a selfmanaged Redis database hosted on ECS

Data Transmission Service (DTS) supports one-way data synchronization between ApsaraDB for Redis instances. This feature is suitable for scenarios such as active geo-redundancy and geo-disaster recovery. This topic describes how to synchronize data from a self-managed Redis database connected over Express Connect, VPN Gateway, or Smart Access Gateway to a self-managed Redis database hosted on Elastic Compute Service (ECS).



#### Prerequisites

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

(?) Note The engine version of the destination Redis database can be 2.8, 3.0, 3.2, 4.0, or 5.0. The engine version of the destination database must be the same as or later than that of the source database. If you 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.
- If the source Redis database is deployed in a cluster architecture, all nodes of the Redis cluster must support the PSYNC command and share the same password.

## Procedure

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

On the buy page, set both Source Instance and Destination 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 Task** in the Actions column.
- 6. Configure the source and destination instances.

1.Configure Source and Destination I	instances in 2.Select Objects to Synchronize		3.Advanced Settings	>	4.	Precheck
Synchronization Task Name:	Redis					
Source Instance Details						
Instance Type:	User-Created Database Connected Over Express Connect, VPN	*				
Instance Region:	Singapore					
* Peer VPC:	vpc-t	-	Proprietary network of Other Apsara Stack Accounts			
Database Type:	Redis					
Instance Mode:	Standalone Oluster					
* IP Address:	172.16.					
* Port Number:	6379					
Database Password:	4	<b>1</b> >				
Destination Instance Details						
Instance Type:	User-Created Database in ECS Instance	Ŧ				
Instance Region:	Singapore					
* ECS Instance ID:	Coloring of Colorest	÷				
Database Type:	Redis					
Instance Mode:	Standalone Oluster					
* Port Number:	6379					
Database Password:		<b>\$</b> >				
					Cancel	Set Whitelist and Next

Section	Parameter	Description
N/A	Synchronizatio n Task Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.
	Instance Type	Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway.
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.

Section	Parameter	Description
	Peer VPC	The ID of the virtual private cloud (VPC) that is connected to the self-managed Redis database.
	Database Type	The value of this parameter is set to <b>Redis</b> .
Source Instance Details	Instance Mode	The architecture of the source Redis database. Select <b>Standalone</b> or <b>Cluster</b> .
	IP Address	The server IP address of the source Redis database.          Image: The server IP address of the server is
		The service port number of the source Redis database. Default value: <b>6379</b> .
	Port Number	<b>Note</b> If the source Redis database is deployed in a cluster architecture, enter the service port number of a master node.
	Database Password	The password of the source Redis database.          ⑦ Note       This parameter is optional and can be left blank if no database password is set.
	Instance Type	Select User-Created Database in ECS Instance.
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.
		The ID of the ECS instance that hosts the destination Redis database.
	Instance ID	<b>Note</b> 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	The architecture of the destination Redis database. Select <b>Standalone</b> or <b>Cluster</b> .
Destination Instance Details		

Section	Parameter	Description
	Port Number	The service port number of the destination Redis database. Default value: <b>6379</b> .
		<b>Note</b> If the destination Redis database is deployed in a cluster architecture, enter the service port number of a master node.
		The password of the destination Redis database.
Database Password	Dat abase Password	<b>Note</b> 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

- You do not need to modify the security settings for ApsaraDB instances (such as and ) and ECS-hosted databases. DTS automatically adds the CIDR blocks of DTS servers to the whitelists of ApsaraDB instances or the security group rules of ECS instances. For more information, see Add the CIDR blocks of DTS servers to the security settings of onpremises databases.
- After data synchronization is complete, we recommend that you remove the CIDR blocks of DTS servers from the whitelists or security groups.

#### 8.

9.

10.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the 🕧 icon next to each failed item

to view details.

- You can troubleshoot the issues based on the causes and run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.

12.

13.

# 11.6. Synchronize data from a selfmanaged Redis cluster to an ApsaraDB for Redis cluster instance

Data Transmission Service (DTS) supports one-way data synchronization between 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 self-managed 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 self-managed Redis cluster. However, you must configure parameters for the source and destination instances based on the actual scenarios.



# Prerequisites

• The database version of the self-managed Redis cluster 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 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

- •
- To ensure the stability of data synchronization, we recommend that you increase the value of the *re pl-backlog-size* parameter in the redis.conf file.
- 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 may become inconsistent between the source and destination databases.
- •
- If an expiration policy is enabled for some keys in the source database, these keys may not be deleted in a timely manner after they expired. Therefore, the number of keys in the destination database may be less than that in the source database. You can run the info command to view the number of keys in the destination database.

**?** Note The number of keys that do not have an expiration policy or have not expired is the same in the source and destination databases.

- •
- •
- Limits on synchronizing a standalone ApsaraDB for Redis instance to an ApsaraDB for Redis cluster instance: Each command can be run only on a single slot in an ApsaraDB for Redis cluster instance. If you perform operations on multiple keys in the source database and the keys belong to different slots, an error occurs.

CROSSSLOT Keys in request don't hash to the same slot

We recommend that you perform operations on only one key during data synchronization. Otherwise, the synchronization task will be interrupted.

#### Procedure

1. Purchase a data synchronization instance.

**On the buy page, set both Source Instance and Destination 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.

1.Configure Source and Destination	n 2.Select Objects to Synchronize	$\rangle$	3.Advanced Settings	$\rightarrow$	4.Precheck
Synchronization Task Name:	Redis				
Source Instance Details					
Instance Type:	User-Created Database in ECS Instance	*			
Instance Region:	Singapore				
* ECS Instance ID:	Construction and Construction	-			
Database Type:	Redis				
Instance Mode: (	Standalone   Cluster				
* Port Number:	6379				
		-			
Database Password:	••••••	<b>()</b>			
Destination Instance Details					
Instance Type:	Redis Instance	*			
Instance Region:	Singapore				
* Instance ID:	r-gs	-			
Database Password:	••••••	<			
				Cancel	Set Whitelist and Next

Section	Parameter	Description				
N/A	Synchronizatio n Task Name	DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.				
	Instance Type	Select User-Created Database in ECS Instance. 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 in ECS Instance as an example. You can also follow the procedure to configure data synchronization tasks for other types of self- managed Redis databases.				
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.				
Source Instance	ECS Instance ID	Select the ID of the Elastic Compute Service (ECS) instance where a master node in the self-managed Redis cluster resides.				
Details	Database Type	The value of this parameter is set to <b>Redis</b> .				
	Instance Mode	Select Cluster.				
	Port Number	Enter the service port number of a master node in the self- managed Redis cluster. In this example, enter <b>7000</b> .				
	Database Password	Enter the password that is used to log on to the self-managed Redis database. <b>Note</b> This parameter is optional and can be left blank if no database password is set.				
	Instance Type	Select Redis Instance.				
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.				
Destination Instance Details	Instance ID	Select the ID of the destination ApsaraDB for Redis cluster instance.				
	Dat abase Password	Enter the database password of the destination ApsaraDB for Redis cluster instance.				

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

**?** Note DTS adds the CIDR blocks of DTS servers to the inbound rule of the source ECS instance and the whitelist of the destination ApsaraDB for Redis cluster instance. This ensures that DTS servers can connect to the source and destination instances.

8.

- 9.
- 10.

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- You can trouble shoot the issues based on the causes and run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.

12.

13.

# 11.7. 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 disks. Based on reliable hot standby architectures and scalable cluster architectures, 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

- An ApsaraDB for Redis instance is created as the destination instance, and the database version of the instance is 2.8, 4.0, or 5.0. For more information, see Step 1: Create an ApsaraDB for Redis instance.
- The available storage capacity 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 until the whole cluster is synchronized.



# Architecture of the Codis cluster

In this topic, a Codis cluster that consists of two codis-groups is used. Each codis-group runs in a master-replica architecture. The following figure shows the architecture of the cluster.

	Group									
demo	New Gro	up Group [1,9999]								
	Add Serv	Data Center	Codis Server Ad	idress to	Group [	1,999	19]			
	GROUPS:	SYNC ALL REPLICA(S): I	ENABLE ALL	EPLICA(S): DISABLE ALL						
	1	Server	Data Center	Master				Memory	Keys	
	SYNC	s 127.0.0.1:6379		NO:ONE		۶	synced	38.05 MB / INF.	db0:keys=31583,expires=0,avg_ttl=0	
	PROMOTE	s 127.0.0.1:6380		127.0.0.1:6379:up		۶	synced	36.61 MB / INF.	db0:keys=31583,expires=0,avg_ttl=0	
	0	Comies	Data Cantar	Mantar				Manani	Kava	
	2	Server	Data Center	Waster		_		memory	Neys	
	SYNC	S 127.0.0.1:6389		NO:ONE		1	synced	38.10 MB / INF.	db0:keys=31636,expires=0,avg_ttl=0	
	DROUGTE	0 107 0 0 1:6300		127.0.0.1:6389:up		F	synced	36.67 MB / INE	db0:keys=31636 expires=0 avg_ttl=0	

## Precautions

- •
- •
- To ensure the stability of data synchronization, we recommend that you increase the value of the *re pl-backlog-size* parameter in the redis.conf file of the source Codis cluster.
- 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 the

destination instance.

- We recommend that you do not run the FLUSHDB or FLUSHALL command in the source Codis cluster. Otherwise, data may become inconsistent between the source and destination databases.
- •
- If an expiration policy is enabled for some keys in the source database, these keys may not be deleted in a timely manner after they expired. Therefore, the number of keys in the destination database may be less than that in the source database. You can run the info command to view the number of keys in the destination database.

**?** Note The number of keys that do not have an expiration policy or have not expired is the same in the source and destination databases.

• The database version of the destination ApsaraDB for Redis instance must 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 synchronize data between different versions of Redis databases, make sure that the versions of the source and destination databases are compatible with each other. For example, you can create a destination pay-as-you-go ApsaraDB for Redis instance to verify the compatibility between the source and destination databases. Then, you can release the instance or change the billing method of the instance to subscription.

#### Commands that can be synchronized

- APPEND
- BIT OP, BLPOP, BRPOP, and BRPOPLPUSH
- DECR, DECRBY, and DEL
- EVAL, EVALSHA, EXEC, EXPIRE, and EXPIREAT
- GEOADD and GET SET
- HDEL, HINCRBY, HINCRBYFLOAT, HMSET, HSET, and HSETNX
- INCR, INCRBY, and INCRBYFLOAT
- LINSERT, LPOP, LPUSH, LPUSHX, LREM, LSET, and LT RIM
- MOVE, MSET, MSET NX, and MULTI
- PERSIST, PEXPIRE, PEXPIREAT, PFADD, PFMERGE, PSETEX, and PUBLISH
- RENAME, RENAMENX, RESTORE, RPOP, RPOPLPUSH, RPUSH, and RPUSHX
- SADD, SDIFFST ORE, SELECT, SET, SET BIT, SET EX, SET NX, SET RANGE, SINTERST ORE, SMOVE, SPOP, SREM, and SUNIONST ORE
- ZADD, ZINCRBY, ZINTERSTORE, ZREM, ZREMRANGEBYLEX, ZUNIONSTORE, ZREMRANGEBYRANK, and ZREMRANGEBYSCORE

? Note

#### Procedure

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

**Note** On the buy page, set Source Instance to **Redis**, set Destination Instance to **Redis**, 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.

1.Configure Source and Des	tination 2.Select (	Objects to Synchronize	>	3.Advanced Settings	$\rangle$	4.Precheck
Synchronization Task	Name: Codic Group1					
Synchronization rask	Cous-Group1					
Source Instance Details						
Instance	Type: User-Created Database in EC	S Instance	•			
Instance R	egion: Singapore					
* ECS Instan	ce ID:		-			
Database	Type: Redis					
Instance	Mode: 🖲 Standalone 🔍 Cluster					
* Port Nu	mber: 6379					
Database Pass	sword:		<b>(</b> \$)			
Destination Instance Details						
	<b>.</b>					
Instance	Redis Instance		v			
Instance R	egion: Singapore					
* Instan	ce ID: r-gs5		•			
Database Pass	sword:		<b>(</b> )			
					Cancel	Set Whitelist and Next
Section	Daramotor	Description				
Section	Parameter	Description				

Parameter	Description
Synchronizatio n Task Name	DTS generates a task name. We recommend that you specify an informative name to identify the task. You do not need to use a unique task name.
Instance Type	Select User-Created Database in ECS Instance.
Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.
	Parameter Synchronizatio n Task Name Instance Type Instance Region

Section	Parameter	Description		
		Select the ID of the Elastic Compute Service (ECS) instance that hosts the master node of the codis-group.		
Source Instance Details	ECS Instance ID	<b>Note</b> DTS synchronizes each codis-group of the Codis cluster by using a data synchronization task until the entire cluster is synchronized. In this step, enter the ECS instance ID of the master node for codis-group 1. When you configure the data synchronization task for codis-group 2, enter the ECS instance ID of the master node for 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 <b>Redis</b> .		
		Select Standalone.		
	Instance Mode	<b>Note</b> You must select <b>Standalone</b> 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 until 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 of the master node.		
		Onte This parameter is optional and can be left blank if no database password is set.		
	Instance Type	Select <b>Redis Instance</b> .		
	Instance Region	The region of the destination cluster. The region is the same as the destination region that you selected on the buy page. The value of this parameter cannot be changed.		
Instance Details	Instance ID	Select the ID of the destination ApsaraDB for Redis instance.		
	Database Password	Enter the database password of the ApsaraDB for Redis instance.		

#### 7.

8. Select the processing mode of conflicting tables and the objects to be synchronized.

#### Data Synchronization Synchronize d

at a bet ween Redis dat abases

1.Configure Source and Destination 2.Select Objects to Synchronize		3.Advanced Settings	$\rightarrow$	4.Precheck
Synchronization Mode:One-Way Synchronization Proccessing Mode In Existed Target Table:    Pre-check and Intercept  Ignore  Available		Selected (To edit an object name or its	s filter haver over the object	and click
If you search globally, please expand the : Q 1 2 3 4 5 6 7 8 9 10 11 12 13 14	> <	Edit.) Learn more.		
Select All		Select All		
*Name batch change:      No      Yes				
			Cance	Previous Next

Parameter	Description		
	DTS synchronizes each codis-group of the Codis cluster in a data synchronization task until the whole cluster is synchronized. When you configure data synchronization for codis-group 1, select <b>Pre-check and</b> <b>Intercept</b> if the ApsaraDB for Redis instance has no data. When you configure data synchronization for codis-groups 2 to N, select <b>Ignore</b> . Otherwise, errors may occur during data synchronization.		
	⑦ Note		
Select the processing mode of conflicting tables	<ul> <li>Pre-check and Intercept: checks whether the destination database is empty. If the destination database is empty, the precheck is passed. If the database is not empty, an error is returned during the precheck and the data synchronization task cannot be started.</li> <li>Ignore: skips the precheck for empty destination databases and continues with data synchronization. If the keys in the synchronization with data synchronization.</li> </ul>		
	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.		
Select the objects to be synchronized	<ul> <li>Select one or more databases from the Available section and click &gt; to move the databases to the Selected section.</li> <li>You can select only databases as the objects to be synchronized. You cannot select keys as the objects to be synchronized.</li> </ul>		

Parameter	Description			
Rename Databases and Tables	In this scenario, you cannot rename objects.			
	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> </ul>			
Replicate Temporary Tables When DMS Performs DDL Operations	<b>Note</b> If online DDL operations generate a large amount of data, the data synchronization task may be delayed.			
	• <b>No</b> : DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.			
	<b>Note</b> If you select No, the tables in the destination database may be locked.			
	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.			
Retry Time for Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.			

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

#### 10. Configure initial synchronization.

1.Configure Source and Destination $ig>$	2.Select Objects to Synchronize	3.Advanced Settings		4.Precheck
Initial Synchronization:Inclu	ide full data + incremental data			
		Cancel	Previous	Save Precheck

**?** Note The value is set to Include full data + incremental data. DTS synchronizes historical data from the source Codis cluster to the destination Redis databases and then synchronizes incremental data.

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

? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
Codis-Group1	Synchronizing	Delay: 1 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task				Total: 1 item(s), Per Page: 20 item(s)	« < 1	> >>

(?) Note You can view the state of the data synchronization task on the Synchronization Tasks page.

14. Create and configure a data synchronization task for the other codis-group by repeating 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.

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
Codis-Group2	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Codis-Group1	Synchronizing	Delay: 2 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task				Total: 2 item(s), Per Page: 20 item(s)	« < 1	> »

In this topic, 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. For more information about how to use DMS to log on to an ApsaraDB for Redis instance, see Use DMS to log on to an ApsaraDB for Redis instance.

ApsaraDB for Redis instance

ts		*	Home							
	✓ Ke	ys: 63220	Instance Info							
v Delete			Version: 4.0.11 Operating Mode: Standalone				Total number of database: 256 Key total: 63200			
уре К	ley Name		Service Listening Port: 6379 Uptime: 0Days 5Hours 32Minutes							
intine k	ey:00000031356	Â	Performance							
TRINC k	ey:00000064287			·						
Jrce C	Codis clust er									
New Grou	up Group [1,9999]									
Add Serv	er Data Center	Codis Server A	ddress to	Group	[1,999	9]				
GROUPS: S	SYNC ALL REPLICA(S	i): ENABLE ALL	REPLICA(S): DISABLE ALL	-						
1	Server	Data Center	Master				Memory	Keys		
SYNC	S 127.0.0.1:6379		NO:ONE		1	synced	38.03 MB / INF.	db0:keys=31584,expires=0,avg_ttl=0 db1:keys=1,expires=0,avg_ttl=0 db15:keys=1,expires=0,avg_ttl=0		
PROMOTE	S 127.0.0.1:6380		127.0.0.1:6379:up		1	synced	36.61 MB / INF.	db0:keys=31584,expires=0,avg_ttl=0 db1:keys=1,expires=0,avg_ttl=0 db15:keys=1,expires=0,avg_ttl=0		
2	Server	Data Center	Master				Memory	Keys		
SYNC	\$ 127.0.0.1:6389		NO:ONE		1	synced	38.06 MB / INF.	db0:keys=31636,expires=0,avg_ttl=0 db1:keys=1,expires=0,avg_ttl=0 db15:keys=1,expires=0,avg_ttl=0		
PROMOTE	s 127.0.0.1:6390		127.0.0.1:6389:up		1	synced	36.67 MB / INF.	db0:keys=31636,expires=0,avg_ttl=0 db1:keys=1,expires=0,avg_ttl=0 db15:keys=1.expires=0.avg_ttl=0		

# 11.8. Synchronize data from a Twemproxy Redis cluster hosted on ECS to an ApsaraDB for Redis instance

ApsaraDB for Redis is a database service that is compatible with the open source Redis protocol and supports a hybrid of memory and disks for storage. Based on reliable hot standby architectures and scalable cluster architectures, 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

- An ApsaraDB for Redis instance is created. For more information, see Step 1: Create an ApsaraDB for Redis instance.
- The available storage capacity of the ApsaraDB for Redis instance is larger than the total size of data stored in the Twemproxy Redis cluster.
- All master nodes in the 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 data of each Redis server in a data synchronization task until the entire 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.



# Procedure

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

**Note** On the buy page, set Source Instance to **Redis**, set Destination Instance to **Redis**, 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.

1.Configure Source and De	estination 2.Select (	Objects to Synchronize	3.Advanced Settings	$\rightarrow$	4.Precheck
Synchronization Task	Name: twemproxy-node1				
Source Instance Details					
Instanc	e Type: User-Created Database in EC	S Instance v			
Instance	Region: Singapore				
* ECS Insta	nce ID:	•			
Databas	e Type: Redis				
Instance	e Mode: 🖲 Standalone 🗌 Cluster				
* Port N	lumber: 6379				
Database Pa	ssword:	4>			
Destination Instance Details					
Instanc	e Type: Redis Instance	T			
Instance	Region: Singapore				
* Insta	nce ID: r-gs5	•			
Database Pa	ssword:	4>			
				Cance	Set Whitelist and Next
Section	Parameter	Description			
N/A	Synchronizatio n Task Name	DTS automatically go you specify an inforr need to use a unique	enerates a task nan native name to ider e task name.	ne. We recon htify the task	nmend that You do not
	Instance Type	Select User-Create	d Database in ECS	6 Instance.	
	Instance	The source region th	at you selected on	the buy pag	e. The value of

N/A	n Task Name	you specify an informative name to identify the task. You do not need to use a unique task name.
	Instance Type	Select User-Created Database in ECS Instance.
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.
		Select the ID of the Elastic Compute Service (ECS) instance that hosts the master node of the Redis server.
	ECS Instance ID	Note DTS synchronizes each Redis server of the Twemproxy Redis cluster by using a data synchronization task until the entire cluster is synchronized. In this step, enter the ECS instance ID for the master node of the Redis server 1. When you configure the data synchronization task for the Redis server 2, enter the ECS instance ID for the master node of the Redis server 2. You can configure data synchronization tasks for all Redis servers by following the procedure described in this topic.

Source Pretince	Parameter	Description			
Details	Database Type	The value of this parameter is set to <b>Redis</b> .			
		Select Standalone.			
	Instance Mode	<b>Note</b> You must select <b>Standalone</b> for this parameter because data from a Twemproxy Redis cluster cannot be synchronized at a time. DTS synchronizes data of each Redis server of the cluster in a data synchronization task until all Redis servers are synchronized.			
	Port Number	Enter the service port number of the master node in the Redis server.			
		Enter the database password of the master node.			
	Database Password	<b>Note</b> This parameter is optional and can be left blank if no database password is set.			
	Instance Type	Select Redis Instance.			
Destination	Instance Region	The region of the destination instance. The region is the same as the destination region that you selected on the buy page. The value of this parameter cannot be changed.			
Instance	Instance ID	Select the ID of the destination ApsaraDB for Redis instance.			
Secure		Enter the database password of the ApsaraDB for Redis instance.			
	Database Password	⑦ Note			

7.

8. Select the processing mode of conflicting tables, and the objects to be synchronized.

#### Dat a Transmission Service

Processing Mode In Existed Target Table:   Pre-check and Intercept Ignore  Available  If you search globally, please expand the  Q  2  2  2  2  2  2  2  2  2  2  2  2			Selected (To edit an object name or i Edit.) Learn more.	ts filter, hover over the o	bject and click
3 4 5 6 7 8 9 7 10 11 11 12 12 13 14	·	> <			
Select All Name batch change:   No Yes			Select All		

Configuration	Description
Select the processing mode of conflicting tables	DTS synchronizes each Redis server of the Twemproxy Redis cluster in a data synchronization task until the entire cluster is synchronized. When you configure data synchronization for the Redis server 1, select <b>Pre-check and Intercept</b> if the ApsaraDB for Redis instance has no data. When you configure data synchronization for the Redis server 2 to N, select <b>Ignore</b> . Otherwise, errors may occur during data synchronization.
	? Note
Select the objects to be synchronized	
Rename Databases and Tables	In this scenario, you cannot rename objects.

Configuration	Description		
Replicate Temporary Tables When DMS Performs DDL Operations	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> </ul>		
	<b>Note</b> If online DDL operations generate a large amount of data, the data synchronization task may be delayed.		
	• <b>No</b> : DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.		
	<b>Note</b> If you select No, the tables in the destination database may be locked.		
	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.		
Retry Time for Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.		

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

1.Configure Source and Destination $\sum$	2.Select Objects to Synchronize	3.Advanced Settings		4.Precheck
Initial Synchronization:Incl	ude full data + incremental data			
		Cancel	Previous	Save Precheck

**?** 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 and then synchronizes incremental data.

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

? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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 initial synchronization is complete and the data synchronization task enters the **Synchronizing** state.

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
twemproxy-node1	Synchronizing	Delay: 0 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task				Total: 1 item(s), Per Page: 20 item(s)	« < 1	> >>

(?) 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 Step 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.

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
dts twemproxy-node2	Synchronizing	Delay: 1 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Restart Task	Switch to Subscription Upgrade More
dts twemproxy-node1	Synchronizing	Delay: 1 Milliseconds Speed: 0TPS(0.00MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task	Switch to Subscription Upgrade More
Pause Task Delete Task				Total: 2 item(s), Per Page:	20 item(s)	

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 instance. The total number of keys is the same as that in the Twemproxy Redis cluster. For more information about DMS usage, see Log on to an ApsaraDB for Redis instance by using DMS.

ApsaraDB for Redis instance

Objects «	Home
DB0 Keys: 63200	Instance Info
New Delete	Version: 4.0.11 Total number of database: 256 Operating Mode: Standalone Key total: 63200
Type Key Name	Service Listening Port: 6379 Uptime: 0Days 5Hours 32Minutes
1 STRING key:00000031356	Performance
2 STRING key:00000064287	

Source Twemproxy Redis cluster

roo	t@			:~#	redis-cli	-p	6379	info grep	db0
db0	:keys	=29421,	expires:	=0,avg_ttl	=0				
roo	t@ ,			:~#	redis-cli	-p	6389	info grep	db0
db0	keys:	<b>=</b> 33779,	expires	=0,avg_ttl	L=0				
roo	t@iZbp	olib0ez	n1xol5w	ofsadZ:~#					

# 12.Synchronize data from an ApsaraDB RDS for SQL Server instance

# 12.1. Synchronize data from an ApsaraDB RDS for SQL Server instance to an AnalyticDB for PostgreSQL instance

This topic describes how to synchronize data from an ApsaraDB RDS for SQL Server instance 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

• An ApsaraDB RDS for SQL Server instance is created. For more information, see Create an ApsaraDB RDS for SQL Server instance and Overview of data synchronization scenarios.

✓ Notice The engine version of the ApsaraDB RDS for SQL Server instance can be 2012, 2014, 2016, 2017, or 2019.

- An AnalyticDB for PostgreSQL instance is created. For more information, see Create an AnalyticDB for PostgreSQL instance.
- The source tables in the ApsaraDB RDS for SQL Server instance contain primary keys.
- The destination tables in the AnalyticDB for PostgreSQL instance contain primary keys or unique indexes.

# Precautions

- DTS uses read and write resources of the source and destination databases during full data migration. This may increase the loads of the database servers. 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 migrate data, evaluate the impact of data migration on the performance of the source and destination databases. We recommend that you migrate data during off-peak hours. For example, you can migrate data when the CPU utilization of the source and destination databases is less than 30%.
- To ensure that the latency of data synchronization is accurate, DTS adds a heartbeat table to the source database. The name of the heartbeat table is dts\_log\_heart\_beat .
- In this scenario, DTS supports schema synchronization for the following types of objects: schema, table, view, function, and procedure.

**Warning** ApsaraDB RDS for SQL Server and AnalyticDB for PostgreSQL are heterogeneous databases. Their data types do not have one-to-one correspondence. We recommend that you evaluate the impact of data type conversion on your business. For more information, see Data type mappings for schema synchronization.

• DTS does not synchronize data of the following types: TIMESTAMP, CURSOR, ROWVERSION, HIERACHYID, SQL\_VARIANT, SPATIAL GEOMETRY, SPATIAL GEOGRAPHY, and TABLE.

# SQL operations that can be synchronized

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

# Permissions required for database accounts

Database	Required permission	Authorization method
ApsaraDB RDS for SQL Server instance	Modify the permissions of a standard account on an ApsaraDB RDS for SQL Server instance	
AnalyticDB for PostgreSQL instance	<ul> <li>The LOGIN permission</li> <li>The SELECT, CREATE, INSERT, UPDATE, and DELETE permissions on the destination tables</li> <li>The CONNECT and CREATE permissions on the destination database</li> <li>The CREATE permission on the destination schemas</li> <li>The COPY permission (the permission to perform memory-based batch copy operations)</li> <li><b>?</b> Note You can use the initial account of the AnalyticDB for PostgreSQL instance.</li> </ul>	Manage users and permissions

# Procedure

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

On the buy page, set Source Instance to SQL Server, Destination Instance to AnalyticDB for PostgreSQL, 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.

1.Select Source and Destination	Instances for 2.Select Object to Be Synchro	nized	3.Precheck
Synchronization Task Name:	ecs sqlserver_to_adbpg sql		
Source Instance Details			
Instance Type:	User-Created Database in ECS Instance		
Instance Region:	China (Hangzhou)		
* ECS Instance ID:	i-bp 🗸		
Database Type:	SQLServer		
* Port Number:			
* Database Account:	dtstest		
* Database Password:	••••••• Ø>		
* Encryption:	● Non-encrypted ○ SSL-encrypted		
Destination Instance Details			
Instance Type:	AnalyticDB for PostgreSQL		
Instance Region:	China (Hangzhou)		
* Instance ID:	gp-		
* Database Name:	dtstestdata		
* Database Account:	dtstest		
* Database Password:	••••••		
			Cancel Set Whitelist and Next

Section	Parameter	Description
N/A	Synchronizati on Task Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.
	Instance Type	Select RDS Instance.
	Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.
	Instance ID	The ID of the source ApsaraDB RDS for SQL Server instance.
	Dat abase Account	The database account of the ApsaraDB RDS for SQL Server instance. For information about the permissions that are required for the account, see Permissions required for database accounts.
	Database Password	The password of the database account.
Source Instance Details		

Section	Parameter	Description			
	Encryption	Select <b>Non-encrypted</b> or <b>SSL-encrypted</b> . If you want to select <b>SSL-encrypted</b> , you must enable SSL encryption for the ApsaraDB RDS for SQL Server instance before you configure the data migration task. For more information, see Configure SSL encryption on an ApsaraDB RDS for SQL Server instance.			
	Licippion	<b>Note</b> The <b>Encryption</b> parameter is available only for regions in the Chinese mainland and the China (Hong Kong) region.			
	Instance Type	The value of this parameter is set to <b>AnalyticDB for</b> <b>PostgreSQL</b> and cannot be changed.			
	Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.			
	Instance ID	The ID of the destination AnalyticDB for PostgreSQL instance.			
Destination Instance Details	Database Name	The name of the destination database.			
	Dat abase Account	The database account of the destination AnalyticDB for PostgreSQL instance. For information about the permissions that are required for the account, see Permissions required for database accounts.			
	Database Password	The password of the database account.			

7.

8. Select the synchronization policy and objects to synchronize.

roccessing Mode In			
kisted Target Table:   Pre-check and Intercept  Ignore			
Merge Multi Tables:O Yes 💿 No Synchronization Type: VInsert VUpdate VDele	te		
Available		Selected (To edit an object name or its filter, hover over the obj Edit.) Learn more.	ject and cli
Expand the tree before you perform a glo		Q	
e Tables		i dtstestdata (20bjects)	
rdscore		dbo.Student	
	>	dbo.customer	
	,		
Select All			
Rename Databases and Tables:       On Not Change Databa	ase and Table Names	Change Database and Table Names	
Retry Time for Failed Connection 720 Minut	tes (?)		
, and a second sec	<u> </u>		

Setting	Description
Initialize Synchronization	<b>Initial Schema Synchronization</b> , <b>Initial Full Data Synchronization</b> , and <b>Initial Incremental Data Synchronization</b> are selected by default. After the precheck is complete, DTS synchronizes the schemas and data of required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization.

Setting	Description				
	• <b>Pre-check and Intercept</b> : 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.				
	<b>Note</b> If the source and destination databases contain identical table names and the tables in the destination database cannot be deleted or renamed, you can use the object name mapping feature to rename the tables that are synchronized to the destination database. For more information, see <b>Rename an object to be synchronized</b> .				
Processing Mode in Existed Target Table	• <b>Ignore Errors and Proceed</b> : skips the precheck for identical table names in the source and destination databases.				
	<ul> <li>Warning If you select Ignore Errors and Proceed, data inconsistency may occur and your business may be exposed to potential risks.</li> <li>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.</li> </ul>				
	If the source and destination databases have different schemas, initial data migration may fail. In this case, only specific columns are migrated, or the data migration task fails.				

Setting Description			
Setting	<ul> <li>Description</li> <li>Yes: In online transaction processing (OLTP) scenarios, sharding is implemented to speed up the response to business tables. However, AnalyticDB for PostgreSQL allows you to store a large amount of data in a single table and makes your SQL queries more efficient. You can merge multiple source tables that have the same schema into a single destination table. This feature allows you to synchronize data from multiple tables in the source database to a single table in AnalyticDB for PostgreSQL.</li> <li>Note <ul> <li>After you select multiple tables from the source database, you must change the names of these tables to the name of the destination table in AnalyticDB for PostgreSQL. To do this, you can use the object name mapping feature. For more information about how to use this feature, see Rename an object to be synchronized.</li> <li>You must add a column nameddts_data_source_ to the destination table in AnalyticDB for PostgreSQL. This column is used to record the data source. The data type of this column is</li> </ul> </li> </ul>		
	used to record the data source. The data type of this column is TEXT. DTS writes column values in the following format: <data synchronization instance ID&gt;:<source database="" name=""/>.<so urce schema name&gt;.<source name="" table=""/> .Such column values allow DTS to identify each source table. For example, dt s*******:dtstestdata.testschema.customer1 indicates that the source table is customer1.</so </data 		
Merge Multi Tables	If you set this parameter to Yes, all the selected source tables in the task are merged into a destination table. If you do not need to merge specific source tables, you can create a separate data synchronization task for these tables.		
	• No: the default value.		

Setting Select the	Description Select the types of operations that you want to synchronize based on your				
operation types	business requirements. All operation types are selected by default.				
Select the objects to be synchronized	Select one or more objects from the <b>Available</b> section and click the icon to add the objects to the <b>Selected</b> section. In this scenario, data synchronization is performed between heterogeneous databases. Therefore, the objects to synchronize are tables, and other objects such as views, triggers and stored procedure are not synchronized to the destination database.				
	<ul> <li>Note</li> <li>By default, after an object is synchronized to the destination instance, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Rename an object to be synchronized.</li> <li>If you set the Merge Multi Tables parameter to Yes, you must change the names of the selected tables to the name of the destination table in the AnalyticDB for PostgreSQL instance. To do this, you can use the object name mapping feature.</li> </ul>				
Add quotation marks to the target object	<ul> <li>Specify whether you need to enclose object names in quotation marks. If you select Yes and the following conditions are met, DTS encloses object names in single quotation marks (') or double quotation marks (") during schema migration and incremental data migration.</li> <li>The business environment of the source database is case-sensitive but the database name contains both uppercase and lowercase letters.</li> <li>A source table name does not start with a letter and contains characters other than letters, digits, and special characters.</li> <li>Note A source table name can contain only the following special characters: underscores (_), number signs (#), and dollar signs (\$).</li> <li>The names of the schemas, tables, or columns that you want to synchronize are keywords, reserved keywords, or invalid characters in the destination database.</li> <li>Note If you select Yes, after DTS synchronizes data to the destination database, you must specify the object name in quotation marks to query the object.</li> </ul>				
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.				

Setting	Description				
Retry Time for Failed Connections	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.				
	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.				

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

					3.Precheck		
Source Database Name	Source Table Name	Type(All) 👻	Primary Key Column	Distribution key	Definition Status(All) 👻		
dtstestdata	customer	Dimension T 🔻	id		Defined		
dtstestdata	Student	Partitioned 1 🔻	StudentID	StudentID <b>v</b>	Defined		
Enter a table name.	Search			Total: 2 item(s), Per Page: 20	▼ item(s) ≪ < 1 > »		
				Cancel	Previous Save Precheck		

**Note** For more information about primary key columns and distribution keys, see **Define** constraints and **Define** table distribution.

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

- ? Note
  - Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
  - If the task fails to pass the precheck, you can click the 🕧 icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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.

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1	> >>

# FAQ

Q: How do I find the tables that are synchronized to the AnalyticDB for PostgreSQL instance?

A: During schema synchronization, DTS synchronizes tables to the destination database based on the schema of the source database. In this example, you can find the customer and Student tables in the dbo schema of the dtstestdata database in the destination instance, as shown in the following figure.



# 12.2. Synchronize data from a selfmanaged SQL Server database hosted on ECS to an AnalyticDB for PostgreSQL instance

This topic describes how to synchronize data from a self-managed SQL Server database that is hosted on Elastic Compute Service (ECS) to an instance by using Data Transmission Service (DTS).

# Prerequisites

• The version of the self-managed SQL Server database is 2008, 2008 R2, 2012, 2014, 2016, or 2017.

**?** Note If you deploy the SQL Server database in an Always On availability group (AOAG), you must use the synchronous-commit mode.

- The tables to be synchronized from the self-managed SQL Server database have primary keys or UNIQUE NOT NULL indexes.
- The available storage space of the instance is larger than the total size of the data in the selfmanaged SQL Server database.

#### Precautions

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization of the source and destination databases is less than 30%.
- To ensure that the data synchronization task runs as expected, do not frequently back up the source database. We recommend that you retain log files for more than three days. Otherwise, you cannot retrieve log files after they are truncated.
- To ensure that the delay time of data synchronization is accurate, DTS adds a heartbeat table to the self-managed SQL Server database. The name of the heartbeat table is Source table name\_dts\_mys ql heartbeat .

## Limits

- DTS does not synchronize the schemas of assemblies, service brokers, full-text indexes, full-text catalogs, distributed schemas, distributed functions, CLR stored procedures, CLR scalar-valued functions, CLR table-valued functions, internal tables, systems, or aggregate functions.
- DTS does not synchronize data of the following types: TIMESTAMP, CURSOR, ROWVERSION, HIERACHYID, SQL\_VARIANT, SPATIAL GEOMETRY, SPATIAL GEOGRAPHY, and TABLE.
- DTS does not synchronize tables that contain computed columns.

# SQL operations that can be synchronized

- Dat a manipulation language (DML) operations: INSERT, UPDATE, and DELETE
- Dat a definition language (DDL) operation: ADD COLUMN

# Permissions required for database accounts

Database	Required permissions	References
Self-managed SQL Server database	The permissions of the sysadmin role	CREATE USER and GRANT (Transact-SQL)

Database	Required permissions	References	
AnalyticDB for PostgreSQL instance	<ul> <li>The LOGIN permission</li> <li>The SELECT, CREATE, INSERT, UPDATE, and DELETE permissions on the destination tables</li> <li>The CONNECT and CREATE permissions on the destination databases</li> <li>The CREATE permission on the destination schemas</li> <li>The COPY permission (the permission to perform memory-based batch copy operations)</li> </ul>	<ul> <li>Create a database account</li> <li>Manage users and permissions</li> </ul>	
	(?) <b>Note</b> You can use the initial account of the AnalyticDB for PostgreSQL instance.		

# Before you begin

Before you configure a data synchronization task, configure log settings and create clustered indexes on the self-managed SQL Server database.

1. Run the following command on the self-managed SQL Server database to change the recovery model to full. You can also change the recovery model by using SQL Server Management Studio (SSMS). For more information, see View or Change the Recovery Model of a Database (SQL Server).

```
use master;
GO
ALTER DATABASE <database_name> SET RECOVERY FULL WITH ROLLBACK IMMEDIATE;
GO
```

#### Parameters:

<database\_name>: the name of the source database. Example:

```
use master;
GO
ALTER DATABASE mytestdata SET RECOVERY FULL WITH ROLLBACK IMMEDIATE;
GO
```

2. Run the following command to create a logical backup for the source database. Skip this step if you have already created a logical backup.

BACKUP DATABASE <database\_name> TO DISK='<physical\_backup\_device\_name>'; GO

Parameters:

- <database\_name>: the name of the source database.
- <physical\_backup\_device\_name>: the storage path and file name of the backup file.

Example:

```
BACKUP DATABASE mytestdata TO DISK='D:\backup\dbdata.bak';
GO
```

3. Run the following command to back up the log entries of the source database:

BACKUP LOG <database\_name> to DISK='<physical\_backup\_device\_name>' WITH init; GO

Parameters:

- <database\_name>: the name of the source database.
- <physical\_backup\_device\_name>: the storage path and file name of the backup file.

Example:

```
BACKUP LOG mytestdata TO DISK='D:\backup\dblog.bak' WITH init;
GO
```

4. Create clustered indexes for the tables that you want to synchronize. For more information, see Create Clustered Indexes.

#### Procedure

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

(?) Note On the buy page, set Source Instance to SQL Server, set Destination 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.
|                           | 1 Colort Course and Docti | nation Instances for             | 2 Calent Object to Do Construction 2 Developed  |  |  |  |
|---------------------------|---------------------------|----------------------------------|---|--|--|--|
|                           | 1.Select Source and Desu  | nation instances for             | 2.select object to be synchronized / 3.precileck  |  |  |  |
|                           | Synchronization Task      | Name: ecs sqlserver_to_adbpg sql | 1   |  |  |  |
|                           |                           |                                  |   |  |  |  |
| Sourc                     | ce Instance Details       |                                  |   |  |  |  |
|                           | Instance                  | Type: User-Created Database in   | ECS Instance 🗸  |  |  |  |
|                           | Instance R                | legion: China (Hangzhou)         |   |  |  |  |
|                           | * ECS Instan              | ice ID: i-bp                     | •   |  |  |  |
|                           | Database                  | Type: SQLServer                  |   |  |  |  |
|                           | * Port Nu                 | umber:                           |   |  |  |  |
|                           | * Database Ac             | count: dtstest                   |   |  |  |  |
|                           | * Database Pas            | sword:                           | Þ   |  |  |  |
|                           | * Encry                   | yption: 🖲 Non-encrypted 🔿 SSL-e  | encrypted   |  |  |  |
|                           |                           |                                  |   |  |  |  |
| Desti                     | nation Instance Details   |                                  |   |  |  |  |
|                           | Instance                  | Type: AnalyticDB for PostgreSQL  |   |  |  |  |
|                           | Instance R                | tegion: China (Hangzhou)         |   |  |  |  |
|                           | * Instan                  | nce ID: gp-                      | -   |  |  |  |
|                           | * Database                | Name: dtstestdata                |   |  |  |  |
| Database Account: dtstest |                           | count: dtstest                   |   |  |  |  |
| * Database Password:      |                           | sword:                           | 4   |  |  |  |
|                           |                           |                                  |   |  |  |  |
|                           |                           |                                  | Cancel Set Whitelist and Next   |  |  |  |
| _                         |                           | _                                |   |  |  |  |
| Sec                       | tion                      | Parameter                        | Description   |  |  |  |
|                           |                           | Cumebro piz ati                  | DTS automatically generates a task name. We recommend that you specify an informative name to identify the task. You do not |  |  |  |
| N/A                       | 4                         | on Task Name                     |   |  |  |  |
|                           |                           | on rask hame                     | need to use a unique task name.   |  |  |  |
|                           |                           |                                  | Select an instance type based on the deployment of the source   |  |  |  |
|                           |                           |                                  | database. In this example, select User-Created Database in  |  |  |  |
|                           |                           |                                  | ECS Instance.   |  |  |  |
|                           |                           | Instance Type                    |   |  |  |  |
|                           |                           | 21                               | <b>ONOTE</b> If you select other instance types, you must   |  |  |  |
|                           |                           |                                  | database. For more information, see Preparation overview.   |  |  |  |
|                           |                           |                                  |   |  |  |  |
|                           |                           |                                  |   |  |  |  |
|                           |                           | Instance                         | The source region that you selected on the buy page. You cannot   |  |  |  |
|                           |                           | Region                           | change the value of this parameter.   |  |  |  |

hosts the source database.

Select the ID of the Elastic Compute Service (ECS) instance that

This parameter is set to **SQLServer** and cannot be changed.

Source Instance

Details

ECS Instance

Database

Туре

ID

Section	Parameter	Description		
	Port Number	Enter the service port number of the source database. The default port number is <b>3306</b> .		
	Dat abase Account	Enter the account of the source database. For information about the permissions that are required for the account, see Permissions required for database accounts.		
	Database Password	Enter the password of the database account.		
		Select Non-encrypted or SSL-encrypted.		
	Encryption	<b>Note</b> The <b>Encryption</b> parameter is available only for regions in the Chinese mainland and the China (Hong Kong) region.		
	Instance Type	The value of this parameter is set to <b>AnalyticDB for</b> <b>PostgreSQL</b> and cannot be changed.		
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.		
	Instance Region Instance ID	The destination region that you selected on the buy page. You cannot change the value of this parameter. Select the ID of the destination AnalyticDB for PostgreSQL instance.		
Destination Instance Details	Instance Region Instance ID Database Name	The destination region that you selected on the buy page. You cannot change the value of this parameter.         Select the ID of the destination AnalyticDB for PostgreSQL instance.         Enter the name of the destination database.		
Destination Instance Details	Instance Region Instance ID Database Name Database Account	<ul> <li>The destination region that you selected on the buy page. You cannot change the value of this parameter.</li> <li>Select the ID of the destination AnalyticDB for PostgreSQL instance.</li> <li>Enter the name of the destination database.</li> <li>Enter the database account of the destination AnalyticDB for PostgreSQL instance. For information about the permissions that are required for the account, see Permissions required for database accounts.</li> </ul>		

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

**?** Note DTS automatically adds the CIDR blocks of DTS servers to the whitelists of the source SQL Server database and the destination AnalyticDB for PostgreSQL instance. You must manually add the CIDR blocks of DTS servers to the inbound rule of the ECS instance that hosts the source database. This ensures that DTS servers can connect to the source and destination instances.

#### 8. Select the synchronization policy and objects to synchronize.

Note: do not clean up the incremental data log generated by the source database after the cleans up the log too early, the DTS incremental task may fail	e DTS task is started when the DTS full task is running. If the source database
Proccessing Mode In Existed Target Table:      Pre-check and Intercept      Ignore	
Merge Multi Tables:O Yes 💿 No	
Synchronization Type: 🗹 Insert 🕼 Update 🕼 Delete	
Available	Selected (To edit an object name or its filter, hover over the object and click Edit.) Learn more.
Expand the tree before you perform a glo	
	dtstestdata (20bjects)
🕂 📂 rdscore	db0.stddent     db0.stddent
>	
<	
Select All	
Select An	Select All
*Rename Databases and Tables:       O Not Change Database and Table Names	Change Database and Table Names
Retry Time for Failed Connection 720 Minutes	
* Add quotation marks to the target 🛛 Yes 💿 No Jbject 🖉	

Setting	Description
Initialize Synchronization	<b>Initial Schema Synchronization, Initial Full Data Synchronization</b> , and <b>Initial Incremental Data Synchronization</b> are selected by default. After the precheck is complete, DTS synchronizes the schemas and data of required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization.

Setting	Description			
	• <b>Pre-check and Intercept</b> : 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.			
Processing Mode in	<b>Note</b> If the source and destination databases contain identical table names and the tables in the destination database cannot be deleted or renamed, you can use the object name mapping feature to rename the tables that are synchronized to the destination database. For more information, see <b>Rename an object to be synchronized</b> .			
Existed Target Table	• <b>Ignore Errors and Proceed</b> : skips the precheck for identical table names in the source and destination databases.			
	<ul> <li>Warning If you select Ignore Errors and Proceed, data inconsistency may occur and your business may be exposed to potential risks.</li> <li>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.</li> <li>If the source and destination databases have different schemas, initial data migration may fail. In this case, only specific columns are migrated, or the data migration task fails.</li> </ul>			

Setting	Description	
Setting	<ul> <li>Yes: In online transaction processing (OLTP) scenarios, sharding is implemented to speed up the response to business tables. However, AnalyticDB for PostgreSQL allows you to store a large amount of data in a single table and makes your SQL queries more efficient. You can merge multiple source tables that have the same schema into a single destination table. This feature allows you to synchronize data from multiple tables in the source database to a single table in AnalyticDB for PostgreSQL.</li> <li>Note         <ul> <li>After you select multiple tables from the source database, you must change the names of these tables to the name of the destination table in AnalyticDB for PostgreSQL. To do this, you can use the object name mapping feature. For more information about how to use this feature, see Rename an object to be synchronized.</li> </ul> </li> </ul>	
Merge Multi T ables	<ul> <li>You must add a column nameddts_data_source to the destination table in AnalyticDB for PostgreSQL. This column is used to record the data source. The data type of this column is TEXT. DTS writes column values in the following format: <data id="" instance="" synchronization="">:<source database="" name=""/>.<source name="" schema=""/>.<source name="" table=""/> . Such column values allow DTS to identify each source table. For example, dt s********:dtstestdata.testschema.customer1 indicates that the source table is customer1.</data></li> <li>If you set this parameter to Yes, all the selected source tables in the task are merged into a destination table. If you do not need to merge specific source tables, you can create a separate data synchronization task for these tables.</li> </ul>	
Select the operation types	Select the types of operations that you want to synchronize based on your business requirements. All operation types are selected by default.	

Setting	Description		
	Select one or more objects from the <b>Available</b> section and click the $i$ icon to add the objects to the <b>Selected</b> section. In this scenario, data synchronization is performed between heterogeneous databases. Therefore, the objects to synchronize are tables, and other objects such as views, triggers and stored procedure are not synchronized to the destination database.		
Select the objects to be synchronized	<ul> <li>Note</li> <li>By default, after an object is synchronized to the destination instance, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Rename an object to be synchronized.</li> <li>If you set the Merge Multi Tables parameter to Yes, you must change the names of the selected tables to the name of the destination table in the AnalyticDB for PostgreSQL instance. To do this, you can use the object name mapping feature.</li> </ul>		
	Specify whether you need to enclose object names in quotation marks. If you select <b>Yes</b> and the following conditions are met, DTS encloses object names in single quotation marks (') or double quotation marks (") during schema migration and incremental data migration.		
	<ul> <li>The business environment of the source database is case-sensitive but the database name contains both uppercase and lowercase letters.</li> <li>A source table name does not start with a letter and contains characters other than letters, digits, and special characters.</li> </ul>		
Add quotation marks to the target object	<b>Note</b> A source table name can contain only the following special characters: underscores (_), number signs (#), and dollar signs (\$).		
	<ul> <li>The names of the schemas, tables, or columns that you want to synchronize are keywords, reserved keywords, or invalid characters in the destination database.</li> </ul>		
	<b>Note</b> If you select Yes, after DTS synchronizes data to the destination database, you must specify the object name in quotation marks to query the object.		
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object nan mapping.		

Setting	Description		
Retry Time for	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.		
Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.		

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

		2.Select Object to Be Synchronized			3.Precheck	
Source Database Name	Source Table Name	Type(All) 👻	Primary Key Column	Distribution key	Definition Status(All) 👻	
dtstestdata	customer	Dimension T 🔻	id		Defined	
dtstestdata	Student	Partitioned 1 V	StudentID	StudentID 🔻	Defined	
Enter a table name.	Search			Total: 2 item(s), Per Page: 20	▼ item(s) ≪ < 1 > ≫	
				Cancel	Previous Save Precheck	

**Note** For more information about primary key columns and distribution keys, see **Define** constraints and **Define** table distribution.

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

- ? Note
  - Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
  - If the task fails to pass the precheck, you can click the 🕡 icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a 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.

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscription Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1 > »

# 13.Synchronize data between PostgreSQL databases

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

This topic describes how to synchronize data from an ApsaraDB RDS for PostgreSQL instance to an 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 synchronize from the ApsaraDB RDS for PostgreSQL instance contain primary keys.
- The destination instance is created. For more information, see Create an instance.

## Precautions

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

ALTER TABLE schema.table REPLICA IDENTITY FULL;

To ensure that the data synchronization task runs as expected, you can perform primary/secondary switchover only on an ApsaraDB RDS for PostgreSQL instance V11. In this case, you must set the rds \_\_failover\_slot\_mode parameter to \_sync . For more information, see Logical Replication Slot Failover.

**Warning** If you perform primary/secondary switchover on a self-managed PostgreSQL database or an ApsaraDB RDS for PostgreSQL instance of other versions, the data synchronization task stops.

# Limits

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

the data synchronization task.

## SQL operations that can be synchronized

INSERT, UPDATE, and DELETE

### Preparations

1. Change the value of the wal\_level parameter for the source RDS instance.

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

- i. Log on to the ApsaraDB RDS console.
- ii. In the top navigation bar, select the region where the RDS instance resides.
- iii. Find the RDS instance and click its ID.
- iv. In the left-side navigation pane, click **Parameters**.
- v. On the **Parameters** page, find the wal\_level parameter and change the parameter value to logical .
- 2. Create a database, schema, and table in the destination instance based on the schema of the objects to synchronize. For more information, see SQL statements.

## Procedure

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

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

- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.
- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination databases.

S

1.Select Source and Destinati	ion Instances for	2.Select Object to Be Synchronized	> 3.Precheck
Synchronization Task N	lame: RDS PostgreSQL_TO_ADB for I	PG	
Source Instance Details			
Instance T	Type: RDS Instance	~	
Instance Re	gion: China (Hangzhou)		
* Instance	e ID: pgm	•	RDS Instances of Other Apsara Stack Accounts
* Database N	ame: dtstestdata		
* Database Acco	ount: dtstest		
* Database Passv	vord:	4>	
Destination Instance Details			
Instance T	Type: AnalyticDB for PostgreSQL		
Instance Re	gion: China (Hangzhou)		
* Instance	e ID: gp-	•	
* Database N	ame: dtstestdata		
* Database Acco	ount: dtstest		
* Database Passv	vord:	4)	
			Cancel Set Whitelist and Next
Section	Parameter	Description	
N/A	Synchronization Task Name	DTS automatically generates a task name. We recommend that you specify an informative name to identify the task. You do not need to use a unique task name.	
	Instance Type	Select RDS Instance	
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.	
	The ID of the instance.	Select the ID of the ApsaraDB RDS for PostgreSQL instanc	
	Database Name	Enter the name of th	e source database.
Source Instance		Enter the account of and the account mus	the AnalyticDB for PostgreSQL instance, t be a privileged account.
Details	Database Account	<b>Note</b> If the source database runs on an ApsaraDB RDS for PostgreSQL instance V9.4 and you synchronize only DML operations, the database account must have the REPLICATION permission.	

S

Section	Parameter	Description		
	Database Password	Enter the password of the source database account.		
	Instance Type	The value of this parameter is set to <b>AnalyticDB for</b> <b>PostgreSQL</b> and cannot be changed.		
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.		
	The ID of the instance.	Select the ID of the instance.		
	Database Name	Enter the name of the destination database.		
Destination Instance Details		<b>Note</b> The database must exist in the instance. Otherwise, you must create a database.		
	Database Account	information, see Create a database account.		
		<b>Note</b> You can also enter an account that has the RDS_SUPERUSER permission. For more information, see Manage users and permissions.		
	Database Password	Enter the password of the 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 ApsaraDB RDS for PostgreSQ and instances. This ensures that DTS servers can connect to the source and destination instances.

8. Select the synchronization policy and the objects to synchronize.

at a bet ween	Post gre SQL dat abase
S	

Proccessing Mode In Existed Target Table:      Pre-check and Intercep	sk may fall ot ○ Clear Target Table ○ Ignoi	re
Synchronization Type: ✓ Insert ✓ Updat Available Expand the tree before you perform a glo   Q	e 🔽 Delete	Selected (To edit an object name or its filter, hover over the object and cl Edit.) Learn more.
<ul> <li>→ Tables</li> <li> + → sys</li></ul>	> <	<ul> <li>customer</li> <li>order</li> </ul>
Select All	of Change Database and Table Nag	Select All Change Database and Table Names

Section	Parameter	Description
Initial synchroniz	Initial synchronization	<b>Initial Full Data Synchronization</b> is selected by default. After the precheck, DTS synchronizes historical data of the required objects from the source instance to the destination instance. The data is the basis for subsequent incremental synchronization.
Specify Source and Destination	Processing Mode In Existed Target Table	<ul> <li>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.     </li> <li>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.     </li> </ul>

	_
	-
	-

Section	Parameter	Description
	Synchronization Type	Select the types of operations that you want to synchronize based on your business requirements.          ⑦ Note       The Alter Table operation is not supported.         • Insert
		<ul> <li>Update</li> <li>Delete</li> <li>AlterTable</li> </ul>
Select Object to be Synchronize d	None	<ul> <li>Select one or more tables from the Available section and click the icon to move the tables to the Selected section.</li> <li>Note <ul> <li>You can select only tables as the objects to synchronize.</li> <li>You can use the object name mapping feature to rename the columns that are synchronized to the destination database. For more information, see Rename an object to be synchronized.</li> </ul> </li> </ul>
Rename Databases and Tables	None	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.

Section	Parameter	Description		
Retry Time	Retry Time for Failed None Connections	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.		
for Failed Connections		<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.		

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

#### ? Note

- Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the 🕧 icon next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.
- 10. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed**. Then, the data synchronization task starts.
- 11. Wait until initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻		Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Subscription	Switch to Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < 1	> >>

# 13.2. Synchronize data from a selfmanaged PostgreSQL database to an AnalyticDB for PostgreSQL instance

This topic describes how to synchronize data from a self-managed PostgreSQL database to an 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 self-managed PostgreSQL database contain primary keys.
- The destination instance is created. For more information, see Create an AnalyticDB for PostgreSQL instance.

## Precautions

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

ALTER TABLE schema.table REPLICA IDENTITY FULL;

• To ensure that the data synchronization task runs as expected, you can perform primary/secondary switchover only on an ApsaraDB RDS for PostgreSQL instance V11. In this case, you must set the rds \_\_failover\_slot\_mode parameter to \_\_sync . For more information, see Logical Replication Slot Failover.

**Warning** If you perform primary/secondary switchover on a self-managed PostgreSQL database or an ApsaraDB RDS for PostgreSQL instance of other versions, the data synchronization task stops.

# Limits

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

## SQL operations that can be synchronized

 $\mathsf{INSERT}$  ,  $\mathsf{UPDATE}$  , and  $\mathsf{DELETE}$ 

# Before you begin

Create a database, schema, and table in the destination instance based on the schema of the objects to be synchronized. For more information, see SQL statements.

#### Procedure

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

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

- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click Data Synchronization.
- 4. At the top of the **Synchronization Tasks** page, select the region where the destination instance resides.
- 5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column.
- 6. Configure the source and destination instances.

1.Select Source and Destinatio	n Instances for	2.Select Object to Be Synchronized	$\rightarrow$	3.Precheck
Synchronization Task Name:	PostgreSQL_TO_ADB for PG			
Source Instance Details				
Instance Type:	User-Created Database in ECS Instan	ce 🗸		
Instance Region:	China (Hangzhou)			
* ECS Instance ID:	i-bp	-		
Database Type:	PostgreSQL			
* Port Number:	5432			
* Database Name:	dtstestdata			
* Database Account:	dtstest			
* Database Password:	•••••	<i>ه</i> >		
Destination Instance Details				
Instance Type:	AnalyticDB for PostgreSQL			
Instance Region:	China (Hangzhou)			
* Instance ID:	gp-	-		
* Database Name:	dtstestdata			
* Database Account:	dtstest			
* Database Password:	•••••	٩>		
				Cancel Set Whitelist and Next
Castian	Deverseter	Description		
Section	Parameter	Description		
N/A	Synchronization Task Name	DTS automatically gene that you specify an info You do not need to use	erates a task na ormative name e a unique task	ame. We recommend for easy identification. name.

Section	Parameter	Description
		Select an instance type based on the deployment of the source database. In this example, select <b>User-Created Database in ECS Instance</b> .
	Instance Type	<b>Note</b> If you select other instance types, you must deploy the network environment for the self-managed database. For more information, see <b>Preparation</b> overview.
		This topic uses <b>User-Created Database in ECS Instance</b> as an example to describe how to configure a data synchronization task.
	Instance Region	The source region that you selected on the buy page. You cannot change the value of this parameter.
Source Instance	ECS Instance ID	Select the ID of the Elastic Compute Service (ECS) instance on which the self-managed PostgreSQL database is deployed.
Details	Database Type	This parameter is set to <b>PostgreSQL</b> and cannot be changed.
	Port Number	Enter the service port number of the self-managed PostgreSQL database.
	Database Name	Enter the name of the source database.
	Database Account	Enter the account of the self-managed PostgreSQL database.
		<b>Note</b> The account must have the permissions of the schema owner.
	Database	
	Password	Enter the password of the database account.
	Instance Type	This parameter is set to <b>AnalyticDB for PostgreSQL</b> and cannot be changed.
	Instance Region	The destination region that you selected on the buy page. You cannot change the value of this parameter.
	Instance ID	Select the ID of the instance.
	Database Name	Enter the name of the destination database.
Destination		

Instance Details

Section	Parameter	Description		
Database Account Database Password		Enter the <b>initial account</b> of the instance. For more information, see Create a database account.		
	Note You can also enter an account that has the RDS_SUPERUSER permission. For more information, see Manage users and permissions.			
	Database Password	Enter the password of the database account.		

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

**?** Note DTS adds the CIDR blocks of DTS servers to the inbound security group rule of the ECS instance and the whitelist of the instance. This ensures that DTS servers can connect to the source and destination instances.

#### 8. Select the synchronization policy and the objects to synchronize.

Initial Synchronization I Initial Full Data Synchronization  Initial Synchronization I Initial Full Data Synchronization  Processing Mode In Excessing Mode In Synchronization Type Insert I Update I Delete   Available	1.Specify Source and Destination Data	bases <b>)</b> 2.Sel	ect Object to Be Sy	nchronized	3.Precheck
Available   Expand the tree before you perform a glo   Image: Tables   Image:	Initial Synchronization: Initial Full D Note: do not clean up the incremental da cleans up the log too early, the DTS incre Proccessing Mode In Existed Target Table: Pre-check ar Synchronization Type: Insert	ata Synchronization ta log generated by the sour mental task may fail nd Intercept O Clear Targe I Update I Delete	rce database after the l t Table  ○ Ignore	DTS task is started when the DTS :	full task is running. If the source database
Select All     Select All       *Rename Databases and Tables: <ul> <li>Do Not Change Database and Table Names</li> <li>Change Database and Table Names</li> <li>* Retry Time for Failed Connection</li> <li>720</li> <li>Minutes ?</li> </ul> <li>Change Database and Table Names</li>	Available Expand the tree before you perform a gl f dtstestdata f ables f sys	0   Q	> <	Selected (To edit an object nan Edit.) Learn more.	ne or its filter, hover over the object and cliv Q s)
* Rename Databases and Tables: <ul> <li>Do Not Change Database and Table Names</li> <li>Change Database and Table Names</li> </ul> * Retry Time for Failed Connection              720	Select All			Select All	
	*Rename Databases and Tables: * Retry Time for Failed Connection	Do Not Change Databa:     720 Minute	se and Table Names	Change Database and Table	e Names

Section	Parameter	Description		
Specify Source and Destination Database	Initial synchronization	<b>Initial Full Data Synchronization</b> is selected by default. After the precheck, DTS synchronizes historical data of the required objects from the source instance to the destination instance. The data is the basis for subsequent incremental synchronization.		
	Processing Mode In Existed Target Table	<ul> <li>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.     <li>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.     </li> </li></ul>		
	Synchroniz at ion Type	Select the types of operations that you want to synchronize based on your business requirements.		
		<b>Note</b> The <b>Alter Table</b> operation is not supported.		
		<ul> <li>Insert</li> <li>Update</li> <li>Delete</li> <li>AlterTable</li> </ul>		
Select Object to be Synchronize d		Select one or more tables from the <b>Available</b> section and click the > icon to move the tables to the <b>Selected</b> section.		
	None	<ul> <li>Note</li> <li>You can select only tables as the objects to synchronize.</li> <li>You can use the object name mapping feature to rename the columns that are synchronized to the destination database. For more information, see Rename an object to be synchronized.</li> </ul>		

Section	Parameter	Description	
Rename Dat abases and T ables	None	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.	
Petry Time	y Time Failed None hections	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.	
for Failed Connections		<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.	
Retry Time for Failed Connections	None	destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails. <b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.	

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

- ? Note
  - Before you can start the data synchronization task, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
  - If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.
- 10. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed**. Then, the data synchronization task starts.
- 11. Wait until initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) <del>-</del>	Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscription Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < <u>1</u> > »

# 14.Synchronize data from a TiDB database

# 14.1. Synchronize data from a selfmanaged TiDB database to an AnalyticDB for MySQL cluster

This topic describes how to synchronize data from a self-managed TiDB database to an cluster by using Data Transmission Service (DTS). In this example, Pump, Drainer, and a Kafka cluster are deployed.

# Prerequisites

- An cluster is created. For more information, see Create an cluster.
- The destination cluster has sufficient storage space.

# Context



The binary log format and implementation mechanism of a TiDB database are different from those of a MySQL database. To synchronize data and minimize modifications to the source TiDB database, you must deploy Pump, Drainer, and a Kafka cluster.

Pump records the binary log files that are generated in TiDB in real time, and sends the binary log files to Drainer. Drainer writes the binary log files to the downstream Kafka cluster. During incremental data synchronization, DTS retrieves data from the Kafka cluster and synchronizes the data to the destination database in real time. For example, DTS can synchronize data to an cluster.

## Precautions

• DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the loads of the database servers. 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 impact of data synchronization on 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 utilization 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 data definition language (DDL) operations on the required objects during data synchronization. Otherwise, data may fail to be synchronized.
- Due to the limits of , if the disk space usage of the nodes in an cluster reaches 80%, the cluster is locked. We recommend that you estimate the required disk space based on the objects that you want to synchronize. You must ensure that the destination cluster has sufficient storage space.
- Prefix indexes cannot be synchronized. If the source database contains prefix indexes, data may fail to be synchronized.

### 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 reported and the data synchronization task is interrupted. 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.

#### Preparations

(?) **Note** The server on which the source database is deployed must be in the same internal network as the servers on which Pump, Drainer, and the Kafka cluster are deployed. This minimizes the impact of network latency on data synchronization.

- 1. Deploy Pump and Drainer. For more information, see TiDB Binlog Cluster Deployment.
- 2. Modify the configuration file of Drainer and specify a Kafka cluster to receive data from Drainer. For more information, see Binlog Slave Client User Guide.
- 3. Deploy a Kafka cluster by using one of the following methods:
  - Deploy a self-managed Kafka cluster. For more information, visit the Apache Kafka official website.

WarningWe recommend that you set themessage.max.bytesandreplica.fetch.max.bytesparameters for the Kafka broker to greater values. We also recommend that youset thefetch.message.max.bytesparameter for the Kafka consumer to a greater value.These settings ensure that the Kafka cluster can receive the binary log files that aregenerated in TiDB. For more information, see Kafka 2.5 Documentation.

• Purchase and deploy a Message Queue for Apache Kafka instance. For more information, see Quick start of Message Queue for Apache Kafka.

**?** Note The Message Queue for Apache Kafka instance must be deployed in the same virtual private cloud (VPC) as the source database server. This ensures reliable data transmission and minimizes the impact of network latency on data synchronization.

- 4. Create a topic in the self-managed Kafka cluster or the Message Queue for Apache Kafka instance.
- 5. Add the CIDR blocks of DTS servers to a whitelist of the TiDB database. For more information, see Add the CIDR blocks of DTS servers to the security settings of on-premises databases.

### Procedure

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

**?** Note On the buy page, set Source Instance to TiDB and set Destination Instance to AnalyticDB for MySQL.

- 2. Log on to the DTS console.
- 3. In the left-side navigation pane, click **Data Synchronization**.
- 4. In the upper part of the **Synchronization Tasks** page, select the region where the data synchronization instance resides.
- 5. Find the data synchronization instance and click **Configure Task** in the Actions column.
- 6. Configure the source and destination instances.
  - i. Configure the task name and the source database.

Synchronization Task Name:	TIDB_T0_ADB	
Source Instance Details		
Instance Type: Instance Region:	User-Created Database in ECS Instance ✓ China (Hangzhou)	
* ECS Instance ID:	i-in it and the spectrum -	
Database Type:	TiDB	
* Port Number:	4000	
* Database Account:	dts	
* Database Password:	••••••	
* Kafka ClusterType:	User-Created Database in ECS Instance $\checkmark$	
* Instance Region:	China (Hangzhou)	
* ECS Instance ID:	Hariban Bagladala 🗸	
* KafkaDort Number		
KaikaPoit Number.	9092	
Kafka Cluster Account:	9092	Optional
Kafka Cluster Account: Kafka Cluster Password	9092	Optional
Kafka Cluster Account: Kafka Cluster Password * Topic:	9092  user_kafka_test  Click Get Topic List and then select the specific topic.	Optional Optional Get Toplic list
Kafka Cluster Account: Kafka Cluster Password * Topic: * Kafka Version	9092  user_kafka_test  Lick Get Topic List and then select the specific topic.  1.0	Optional Optional Get Toplic list

Parameter	Description	
Synchronization Task Name	The task name that DTS automatically generates. We recommend that you specify a descriptive name that makes it easy to identify the task. You do not need to use a unique task name.	
	The access method of the source database. In this example, <b>User-Created Database in ECS Instance</b> is selected.	
Instance Type	<b>Note</b> If you select other instance types, you must deploy the network environment for the self-managed database. For more information, see <b>Preparation overview</b> .	
Instance Region	The source region that you selected on the buy page. The value of this parameter cannot be changed.	
Database Type	The value of this parameter is set to <b>TiDB</b> and cannot be changed.	

Parameter	Description		
Port Number	The service port number of the source TiDB database. Default value: 4000.		
Database Account	The account of the source TiDB database. The account must have the SELECT permission on the objects to synchronize and the SHOW VIEW permission. For more information, see Privilege Management.		
Database Password	The password of the database account.		
	The access method of the Kafka cluster. In this example, <b>User-Created</b> <b>Database in ECS Instance</b> is selected. If the Kafka cluster is connected over other methods, you must deploy the network environment for the Kafka cluster. For more information, see <b>Preparation overview</b> .		
Kafka Cluster Type	Note You cannot select Message Queue for Apache Kafka for the Kafka Cluster Type parameter. If you deploy a Message Queue for Apache Kafka instance, you must select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway. Then, you must select the VPC to which the Message Queue for Apache Kafka instance belongs.		
Instance Region	The value of this parameter is the same as the region of the source database and cannot be changed.		
ECS Instance ID	The ID of the Elastic Compute Service (ECS) instance that hosts the self- managed Kafka cluster.		
Kafka Port Number	The service port number of the self-managed Kafka cluster. Default value: 9092.		
Kafka Cluster Account	The username that is used to log on to the Kafka cluster. If no authentication is enabled for the Kafka cluster, you do not need to enter the username.		
Kafka Cluster Password	The password that corresponds to the username. If no authentication is enabled for the Kafka cluster, you do not need to enter the password.		
Торіс	Click <b>Get Topic List</b> and select a topic name from the drop-down list.		
Kafka Version	The version of the self-managed Kafka cluster.		
Kafka Cluster Encryption	Select <b>Non-encrypted</b> or <b>SCRAM-SHA-256</b> based on your business and security requirements.		

## ii. Configure the destination database.

Destination Instance Details	
Instance Type:	AnalyticDB
Instance Region:	China (Hangzhou)
*Version:	• 3.0
* Database:	am-
* Database Account:	dtstest
* Database Password:	······ <

Parameter	Description
Instance Type	The value of this parameter is set to <b>AnalyticDB</b> and cannot be changed.
Instance Region	The destination region that you selected on the buy page. The value of this parameter cannot be changed.
Version	The value of this parameter is set to <b>3.0</b> and cannot be changed.
Database	The ID of the destination cluster.
Database Account	The database account of the destination cluster. The account must have read and write permissions on the destination database. For more information, see Create a database account.
Database Password	The password of the database account.

#### 7.

#### 8. Select the synchronization policy and the objects to synchronize.

Setting	Description
Select the initial synchronization types	You must select both <b>Initial Schema Synchronization</b> and <b>Initial Full Data</b> <b>Synchronization</b> in most cases. After the precheck is complete, DTS synchronizes the schemas and data of required objects from the source instance to the destination cluster. The schemas and data are the basis for subsequent incremental synchronization.

Setting	Description		
Select the processing mode of conflicting tables	<ul> <li>Precheck and Report Errors: checks whether the destination 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.</li> <li>Note You can use the object name mapping feature to rename the tables that are synchronized to the destination database. If the source and destination database cannot be deleted or renamed, you can use this feature. For more information, see Rename an object to be synchronized.</li> <li>Ignore Errors and Proceed: skips the precheck for identical table names in the source and destination databases.</li> <li>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.</li> <li>If the source and destination databases have different schemas, initial data synchronized, or the data next the same schema, DTS does not synchronized, or the data records that have the same primary keys as data records in the database.</li> </ul>		
Specify whether to merge tables	<ul> <li>If you select Yes, DTS adds thedts_data_source column to each table to store data sources. In this case, DDL operations cannot be synchronized.</li> <li>No is selected by default. In this case, DDL operations can be synchronized.</li> <li>Note If you set this parameter to Yes, all the selected source tables in the task are merged into the destination table. To merge only the data source columns of specific tables, you can create two data synchronization tasks.</li> </ul>		
Select the operation types to synchronize	Select the types of operations that you want to synchronize based on your business requirements. All operation types are selected by default. For more information, see SQL operations that can be synchronized.		

Setting	Description		
	Select one or more objects from the <b>Available</b> section and click the > icon to add the objects to the <b>Selected</b> section. You can select tables or databases as the objects to synchronize.		
Select the objects to synchronize	<ul> <li>Note</li> <li>If you select a database as the object to synchronize, all schema changes in the database are synchronized to the destination database.</li> <li>If you select a table as the object to synchronize, only the ADD COLUMN operations that are performed on the table are synchronized to the destination database.</li> <li>By default, after an object is synchronized to the destination cluster, the name of the object remains unchanged. You can use the object name mapping feature to rename the objects that are synchronized to the destination cluster. For more information, see Rename an object to be synchronized.</li> </ul>		
Rename Databases and Tables	You can use the object name mapping feature to rename the objects that are synchronized to the destination instance. For more information, see Object name mapping.		
Replicate	<ul> <li>If you use Data Management (DMS) to perform online DDL operations on the source database, you can specify whether to synchronize temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> <li>Yes: DTS synchronizes the data of temporary tables generated by online DDL operations.</li> <li>Note If online DDL operations generate a large amount of data,</li> </ul>		
Temporary Tables When DMS	the data synchronization task may be delayed.		
Performs DDL Operations	• <b>No</b> : DTS does not synchronize the data of temporary tables generated by online DDL operations. Only the original DDL data of the source database is synchronized.		
	<b>Note</b> If you select No, the tables in the destination database may be locked.		

Setting	Description
Retry Time for	By default, if DTS fails to connect to the source or destination database, DTS retries within the next 720 minutes (12 hours). You can specify the retry time based on your needs. If DTS reconnects to the source and destination databases within the specified time, DTS resumes the data synchronization task. Otherwise, the data synchronization task fails.
Failed Connections	<b>Note</b> When DTS retries a connection, you are charged for the DTS instance. We recommend that you specify the retry time based on your business needs. You can also release the DTS instance at your earliest opportunity after the source and destination instances are released.

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

10. Specify a type for the tables that you want to synchronize to the destination database.

	Destination Instances > 2.Au	thorize AnalyticDB Account	3.Select Objects to	Synchronize	
AnalyticDB Table Group	AnalyticDB Table Name	Type(All) 👻	Primary Key Column	Distribution Column	Definition Status(All) 👻
tidbtestdata	customer	Partitioned 1 🔻	id	id 🔻	Defined
tidbtestdata	order	Partitioned 1 🔻	orderid	orderid 🔻	Defined
Set All to Partitioned Table	Set All to Dimension Table Enter a table n	ame. Search		Total: 2 item(s), Per Page: 2	0 ▼ item(s) ≪ < 1 > »
				Cancel	Previous Save Precheck

Onte After you select Initial Schema Synchronization, you must specify the type, primary key column, and partition key column for the tables that you want to synchronize to the destination cluster. 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, DTS performs a precheck. You can start the data synchronization task only after the task passes the precheck.
- If the task fails to pass the precheck, you can click the next to each failed item

to view details.

- After you troubleshoot the issues based on the causes, you can run a precheck again.
- If you do not need to troubleshoot the issues, you can ignore failed items and run a precheck again.
- 12. Close the **Precheck** dialog box after the following message is displayed: **Precheck Passed**. Then, the data synchronization task starts.
- 13. Wait until initial synchronization is completed and the data synchronization task enters the **Synchronizing** state.

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

Instance ID/Task Name	Status	Synchronization Details	Billing Method	Synchronization Mode(All) 👻	Actions
0	Synchronizing	Delay: 0 Milliseconds Speed: 0.00RPS/(0.000MB/s)	Pay-As-You-Go	One-Way Synchronization	Pause Task Switch to Subscription Upgrade More
Pause Task Delete Task			Total: 1 item(s),	Per Page: 20 item(s)	« < <u>1</u> > »