ALIBABA CLOUD

Alibaba Cloud

Data Transmission Service Introduction

Document Version: 20220712

Alibaba Cloud

Legal disclaimer

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

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

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

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

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

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

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

Document conventions

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

Table of Contents

1.What is Data Transmission Service?

Alibaba Cloud Data Transmission Service (DTS) is a real-time data streaming service. DTS supports data transmission between data sources such as relational, NoSQL, and online analytical processing (OLAP) databases. DTS provides the data synchronization, data migration, change tracking, data integration, and data processing features. This enables you to manage data within a secure, scalable, and high-availability architecture.

Benefits

Compared with traditional data migration and synchronization tools, DTS has the following advantages: high compatibility, high performance, security, reliability, and ease of use. This allows you to simplify data transmission and focus on your application development.

- **Ease of use**

  The DTS console is a visual management interface that provides a wizard-like process for creating DTS tasks.

- **Diverse features**
  - DTS allows you to migrate or synchronize data between heterogeneous databases that run on different database engines and have different architectures.
  - DTS provides multiple data replication modes, including data migration, data synchronization, data integration, and change tracking.

- **High performance**
  - DTS uses servers with high specifications to ensure the performance of each data synchronization or migration instance.
  - DTS has its core infrastructure optimized and can achieve a peak data transmission rate of 70 MB/s.
  - DTS supports concurrent compressed data transmission that minimizes the bandwidth utilization.

- **High security and reliability**
  - DTS supports disaster recovery of nodes. A failing task connection can be recovered within seconds.
  - DTS supports resumable transmission to resolve transmission interruptions caused by exceptions such as hardware and network failures.
  - DTS allows you to manage access to DTS tasks with fine-grained custom policies created by using Resource Access Management (RAM).
  - DTS supports data verification to ensure data integrity and consistency between the source and destination databases.

For more information about DTS, see Benefits and Scenarios.

Features

Feature	Description	References
Data synchronization	You can use DTS to synchronize data between data sources in real time. This feature is suitable for the following scenarios: active geo-redundancy, geo-disaster recovery, zone-disaster recovery, cross-border data synchronization, cloud-based business intelligence (BI) systems, and real-time data warehousing.	• Overview of data synchronization scenarios • Data synchronization mode
Data migration	You can use DTS to migrate data between homogeneous and heterogeneous data sources. This feature is suitable for the following scenarios: data migration to Alibaba Cloud, data migration between databases within Alibaba Cloud, and database splitting and scale-out. **Note** Data migration can partly achieve data synchronization in specific scenarios. However, data migration and data synchronization differ in applicable scenarios, functionality, and billing. For more information about the differences between data migration and data synchronization, see the "What are the differences between data migration and data synchronization?" section of the FAQ topic.	• Overview of data migration scenarios • Data migration mode
Data integration	Data integration is an advanced feature on top of data migration. You can configure a scheduling policy to migrate schemas and historical data from a source database to a destination database on a regular basis. The data integration feature allows you to flexibly build a data warehouse such as an offline data warehouse.	Data integration mode
Change tracking	You can use DTS to track incremental data from self-managed MySQL databases, ApsaraDB RDS for MySQL instances, PolarDB for MySQL clusters, PolarDB-X instances, and self-managed Oracle databases in real time. Then, you can consume the tracked data as needed. This feature is suitable for the following scenarios: cache updates, business decoupling and asynchronous data processing, real-time data synchronization between heterogeneous databases, and real-time data synchronization that involves extract, transform, load (ETL) operations.	Change tracking mode
Data processing	You can use DTS to extract, transform, process, and load streaming data. This feature helps enterprises navigate real-time data processing and computing scenarios and empower enterprises to go digital.	What is ETL?What is ETL?

Supported databases

DTS supports a variety of data sources. For more information, see Supported sources and targets.

Management tools

You can use the following methods to create and configure DTS tasks and perform task O&M:

- DTS console: a web service page used to efficiently manage DTS tasks.
- API: provides API operations that you can call to perform all operations that are available in the DTS console.
- SDK: provides SDKs for a variety of programming languages to perform all operations that are available in the DTS console.

Pricing

For more information, see Billing methods.

Related Alibaba Cloud services

- Elastic Compute Service (ECS): Self-managed databases hosted on ECS instances can serve as the data sources or destinations for data migration, data synchronization, and change tracking tasks.
- ApsaraDB RDS: ApsaraDB RDS is a stable, reliable, and scalable online database service. ApsaraDB RDS provides a variety of solutions for disaster recovery, backup, restoration, and migration to facilitate database O&M.
- PolarDB: PolarDB is a next-generation relational database service that is developed by Alibaba Cloud. PolarDB is compatible with the MySQL, PostgreSQL, and Oracle database engines. PolarDB provides superior performance in storage and computing to meet the requirements of enterprises. Each PolarDB cluster provides a storage capacity of up to 100 TB and supports up to 16 nodes.
- PolarDB-X: PolarDB-X is a cloud-based distributed database service that is designed to handle the bottlenecks of standalone databases. PolarDB-X is developed in-house by Alibaba Cloud. PolarDB-X is highly compatible with MySQL protocols and syntax and supports automatic sharding, online smooth scaling, elastic scaling, and read/write splitting. PolarDB-X provides O&M capabilities throughout the lifecycle of databases.
- ApsaraDB for Redis: ApsaraDB for Redis is an online key-value storage database service that provides high performance and is compatible with open source Redis protocols.
- ApsaraDB for MongoDB: ApsaraDB for MongoDB is a MongoDB-compatible database service that is developed based on the Apsara distributed operating system and a high-reliability storage engine. ApsaraDB for MongoDB uses a multi-node architecture to ensure high availability, and supports elastic scaling, disaster recovery, backup and restoration, and performance optimization.

2.Supported sources and targets

This topic describes the supported data migration paths (source-target pairs) and replication features of all replication modes. For each migration path, the supported cloud-based and user-created databases and their corresponding replication features are listed.

Data migration

The data migration mode migrates your data between data stores. This mode is typically used for one-time migrations.

Supported user-created deployments

A user-created source or target database, such as a MySQL, SQL Server, or Oracle database, can be based on one of the following hosting and network deployments:

- User-created database with public IP address
- Database without public IP:Port (Accessed through database gateway)
- User-created database accessed through Cloud Enterprise Network(CEN)
- User-created database hosted on ECS Instance
- User-created database connected over Express Connect, VPN Gateway, or Smart Access Gateway

Supported migration paths:

Source database	Target database	Migration phases
	User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0	• Schema migration • Full data migration • Incremental data migration
	ApsaraDB RDS for MySQL All versions	• Schema migration • Full data migration • Incremental data migration

Source database	Target database	Migration phases
- User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0 - ApsaraDB RDS for MySQL All versions	Apsara PolarDB for MySQL All versions	- Schema migration - Full data migration - Incremental data migration
	Distributed Relational Database Service (DRDS) MySQL 5.x ⓘ **Note** Apsara PolarDB for MySQL clusters are not supported.	- Full data migration - Incremental data migration
	AnalyticDB for MySQL 2.0 and 3.0	- Schema migration - Full data migration - Incremental data migration
	User-created PostgreSQL database 9.4, 9.5, 9.6, 10.x, 11.x, and 12	- Full data migration - Incremental data migration
	User-created Oracle database (RAC and non-RAC deployments) 9i, 10g, 11g, 12c, 18c, and 19c	- Schema migration - Full data migration - Incremental data migration

Source database	Target database	Migration phases
ApsaraDB for MariaDB TX 10.3	ApsaraDB for MariaDB TX 10.3	• Schema migration • Full data migration • Incremental data migration
	ApsaraDB RDS for MySQL All versions	• Schema migration • Full data migration • Incremental data migration
	User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0	• Schema migration • Full data migration • Incremental data migration
Apsara PolarDB for MySQL All versions	Apsara PolarDB for MySQL All versions	• Schema migration • Full data migration • Incremental data migration
	User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0	• Schema migration • Full data migration • Incremental data migration

Source database	Target database	Migration phases
	ApsaraDB RDS for MySQL All versions	- Schema migration - Full data migration - Incremental data migration
	AnalyticDB for MySQL 2.0 and 3.0	- Schema migration - Full data migration - Incremental data migration
Apsara PolarDB O Edition All versions	Apsara PolarDB O Edition All versions	- Schema migration - Full data migration - Incremental data migration
	User-created Oracle database (RAC and non-RAC deployments) 9i, 10g, 11g, 12c, 18c, and 19c	- Schema migration - Full data migration - Incremental data migration

Source database	Target database	Migration phases
- User-created SQL Server database 2005, 2008, 2008 R2, 2012, 2014, 2016, and 2017 **Note** ○ SQL Server Cluster and SQL Server Always On availability groups are not supported. ○ If the version of the source database is 2005, incremental data migration is not supported. - ApsaraDB RDS for SQL Server 2008, 2008 R2, 2012, 2014, 2016, and 2017 **Note** If the version of the source database is 2008 or 2008 R2, incremental data migration is not supported.	User-created SQL Server database 2005, 2008, 2008 R2, 2012, 2014, 2016, and 2017 **Note** SQL Server Cluster and SQL Server Always On availability groups are not supported.	- Schema migration - Full data migration - Incremental data migration
	ApsaraDB RDS for SQL Server 2008, 2008 R2, 2012, 2014, 2016, and 2017	- Schema migration - Full data migration - Incremental data migration
	User-created Oracle database (RAC and non-RAC deployments) 9i, 10g, 11g, 12c, 18c, and 19c	- Schema migration - Full data migration - Incremental data migration
	ApsaraDB for PolarDB O Edition All versions	- Schema migration - Full data migration - Incremental data migration

Source database	Target database	Migration phases
User-created Oracle database (RAC and non-RAC deployments) 9i, 10g, 11g, 12c, 18c, and 19c	ApsaraDB RDS for PPAS 9.3 and 10	● Schema migration ● Full data migration ● Incremental data migration
	User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0	● Schema migration ● Full data migration ● Incremental data migration
	ApsaraDB RDS for MySQL All versions	● Schema migration ● Full data migration ● Incremental data migration
	Distributed Relational Database Service (DRDS) MySQL 5.x ? **Note** Apsara PolarDB for MySQL clusters are not supported.	● Full data migration ● Incremental data migration
	AnalyticDB for MySQL 2.0 and 3.0	● Schema migration ● Full data migration ● Incremental data migration

Source database	Target database	Migration phases
- User-created PostgreSQL database 9.4, 9.5, 9.6, 10.x, 11.x, and 12 - ApsaraDB RDS for PostgreSQL 9.4, 10, 11, and 12	- User-created PostgreSQL database 9.4, 9.5, 9.6, 10.x, 11.x, and 12 - ApsaraDB RDS for PostgreSQL 9.4, 10, 11, and 12	- Schema migration - Full data migration - Incremental data migration
- User-created MongoDB database (single-node, replica set, or sharded cluster deployment) 3.0, 3.2, 3.4, 3.6, 4.0, and 4.2 - ApsaraDB for MongoDB instance (single-node or replica set deployment) All versions	- User-created MongoDB database (single-node, replica set, or sharded cluster deployment) 3.0, 3.2, 3.4, 3.6, 4.0, and 4.2 - ApsaraDB for MongoDB instance (single-node, replica set, or sharded cluster deployment) All versions	- Full data migration - Incremental data migration **Note** As a NoSQL database, MongoDB does not require schema migration.
User-created Redis database (single-node deployment) 2.8, 3.0, 3.2, 4.0, and 5.0	User-created Redis database (single-node and cluster deployment) 2.8, 3.0, 3.2, 4.0, and 5.0	- Full data migration - Incremental data migration **Note** As a NoSQL database, Redis does not require schema migration.
	ApsaraDB for Redis instance (single-node and cluster deployment) Community versions 4.0 and 5.0	- Full data migration - Incremental data migration

Source database	Target database	Migration phases
User-created TiDB database	User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0	• Schema migration • Full data migration • Incremental data migration
	ApsaraDB RDS for MySQL All versions	• Schema migration • Full data migration • Incremental data migration
	Apsara PolarDB for MySQL All versions	• Schema migration • Full data migration • Incremental data migration
User-created Db2 database 9.7 and 10.5	User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0	• Schema migration • Full data migration • Incremental data migration

Data synchronization

The data synchronization mode replicates ongoing updates between data stores. This mode is typically used for replication between server nodes in a highly available system based on a redundant and distributed deployment. This mode is also used for OLTP-to-OLAP replications to allow for real-time analytics.

Supported user-created deployments:

A user-created source or target database, such as a MySQL or Redis database, can be based on one of the following hosting and network deployments:

- User-created database hosted on ECS Instance
- User-created database connected over Express Connect, VPN Gateway, or Smart Access Gateway
- Database without public IP:Port (Accessed through database gateway)

- User-created database accessed through Cloud Enterprise Network (CEN)

Supported replication pairs:

Source database	Target database	Initial data load type	Synchronization directionality
• User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0 • ApsaraDB RDS for MySQL All versions	User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0	Initial schema synchronization Initial full data synchronization	One-way replication Two-way replication
	ApsaraDB RDS for MySQL All versions	Initial schema synchronization Initial full data synchronization	One-way replication Two-way replication
	Apsara PolarDB for MySQL All versions	Initial schema synchronization Initial full data synchronization	One-way replication
	AnalyticDB for MySQL 2.0 and 3.0	Initial schema synchronization Initial full data synchronization	One-way replication
	AnalyticDB for PostgreSQL (formerly known as HybridDB for PostgreSQL) 4.3 and 6.0	Initial schema synchronization Initial full data synchronization	One-way replication
	Elasticsearch 5.5, 6.3, 6.7, and 7.4	Initial schema synchronization Initial full data synchronization	One-way replication
	MaxCompute All versions	Initial schema synchronization Initial full data synchronization	One-way synchronization

Source database	Target database	Initial data load type	Synchronization directionality
	User-created Kafka database Cluster versions 0.10 and 1.0	Initial schema synchronization Initial full data synchronization	One-way replication
DRDS MySQL 5.x **Note** ApsaraDB RDS for MySQL instances (MySQL 8.0) or Apsara PolarDB for MySQL clusters are not supported	DRDS MySQL 5.x **Note** ApsaraDB RDS for MySQL instances (MySQL 8.0) or Apsara PolarDB for MySQL clusters are not supported	Initial full data synchronization	One-way replication
	AnalyticDB for MySQL 2.0 and 3.0	Initial schema synchronization Initial full data synchronization	One-way replication
	AnalyticDB for PostgreSQL (Previous name: HybridDB for PostgreSQL) 4.3 and 6.0	Initial schema synchronization Initial full data synchronization	One-way replication
	User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0	Initial schema synchronization Initial full data synchronization	One-way replication
	ApsaraDB RDS for MySQL All versions	Initial schema synchronization Initial full data synchronization	One-way replication
	Apsara PolarDB for MySQL All versions	Initial schema synchronization Initial full data synchronization	One-way replication

Source database Apsara PolarDB for MySQL	Target database	Initial data load type	Synchronization directionality
All versions	AnalyticDB for MySQL 2.0 and 3.0	Initial schema synchronization Initial full data synchronization	One-way replication
	User-created Kafka database Cluster versions 0.10 and 1.0	Initial schema synchronization Initial full data synchronization	One-way replication
	Elasticsearch 5.5, 6.3, and 6.7	Initial schema synchronization Initial full data synchronization	One-way replication
	MaxCompute All versions	Initial schema synchronization Initial full data synchronization	One-way replication
• ApsaraDB RDS for PostgreSQL 9.4, 10, 11, and 12 • User-created PostgreSQL database 9.x to 12	AnalyticDB for PostgreSQL (formerly known as HybridDB for PostgreSQL) 4.3 and 6.0	Initial full data synchronization	One-way replication

Source database	Target database	Initial data load type	Synchronization directionality
- User-created Redis database (single-node and cluster deployment) 2.8, 3.0, 3.2, 4.0, and 5.0 - ApsaraDB for Redis instance (standard, read/write splitting, and cluster instances) Community versions 4.0 and 5.0 - ApsaraDB for Redis Enhanced Edition instance (standard, read/write splitting, and cluster instances) 5.0	- User-created Redis database (single-node or cluster deployment) 2.8, 3.0, 3.2, 4.0, and 5.0 - ApsaraDB for Redis instance (standard, read/write splitting, and cluster instances) Community versions 4.0 and 5.0 - ApsaraDB for Redis Enhanced Edition instance (standard, read/write splitting, and cluster instances) 5.0	Initial full data synchronization **Note** Redis is a NoSQL database that does not require initial schema synchronization.	One-way replication Two-way replication **Note** Only ApsaraDB for Redis Enhanced Edition instance (5.0) support two-way synchronization

Change tracking

The change tracking mode captures data updates and expose them as a publish/subscribe stream of updates. This mode is typically used for systems that require asynchronous replication for better performance over mission-critical workloads.

Supported user-created deployments:

A user-created source database, such as MySQL, can be based on one of the following hosting and network deployments:

- User-created database hosted on ECS Instance
- User-created database connected over Express Connect, VPN Gateway, or Smart Access Gateway
- Database without public IP:Port (Accessed through database gateway)
- User-created database with public IP address

Source database	Data update types
User-created MySQL database 5.1, 5.5, 5.6, 5.7, and 8.0	• Data updates • Schema updates
ApsaraDB RDS for MySQL All versions	
User-created Oracle database 9i, 10g, and 11g	
PolarDB MySQL 5.6, 6.7, 8.0	

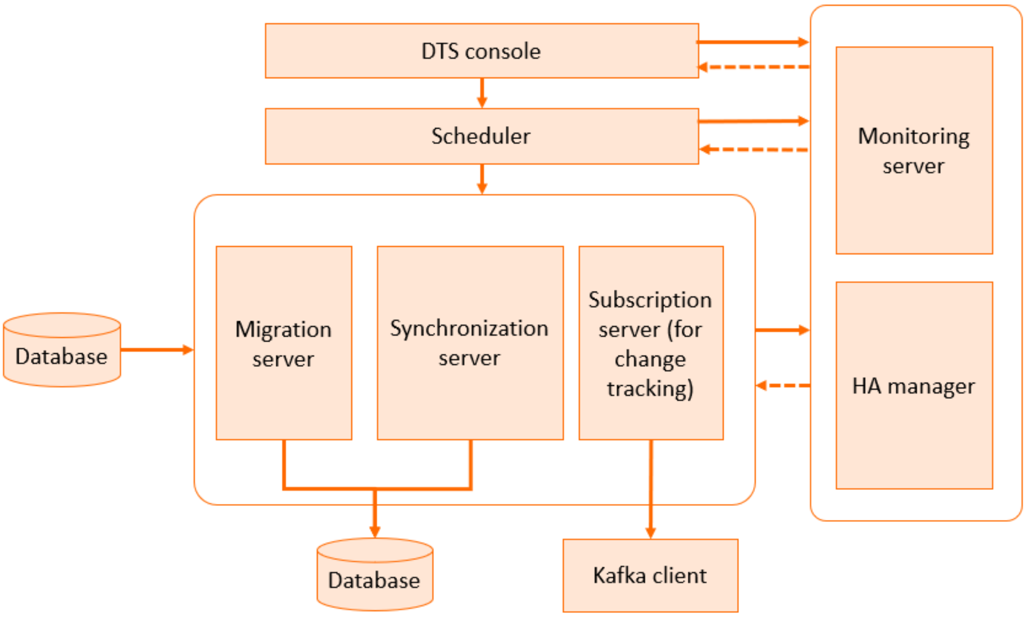
3.Service architecture

This topic describes the architecture of Data Transmission Service (DTS) and how it works in each replication mode.

DTS architecture

The architecture is illustrated in the following figure:

DTS architecture



The architecture of Data Transmission Service provides the following features:

- Primary/secondary redundancy:

  Each function of DTS is deployed on multiple servers with primary/secondary redundancy. The HA manager continuously performs health checks on each server. If one server functions abnormally, the workloads on that server are switched to a healthy server with minimal delay.

- Endpoint change detection:

  For continuous data replications, such as data synchronization and change tracking, the HA manager detects the changes made to the endpoints of source ApsaraDB instances. If an instance endpoint has changed, the HA manager reconfigures the data source to keep the data connection functioning.

How DTS works in data migration mode

A data migration task consists of several phases, namely schema migration, full data migration, and incremental data migration. To keep the source data operational during migration, you must select all of these phases in the migration task configuration wizard.

Schema migration

Before migrating data, DTS needs to re-create the schema in the target database. For heterogeneous migrations, DTS parses the data definition language (DDL) code of the source database, translates the code into the syntax of the target database, and then re-creates the schema objects in the target database.

Full data migration

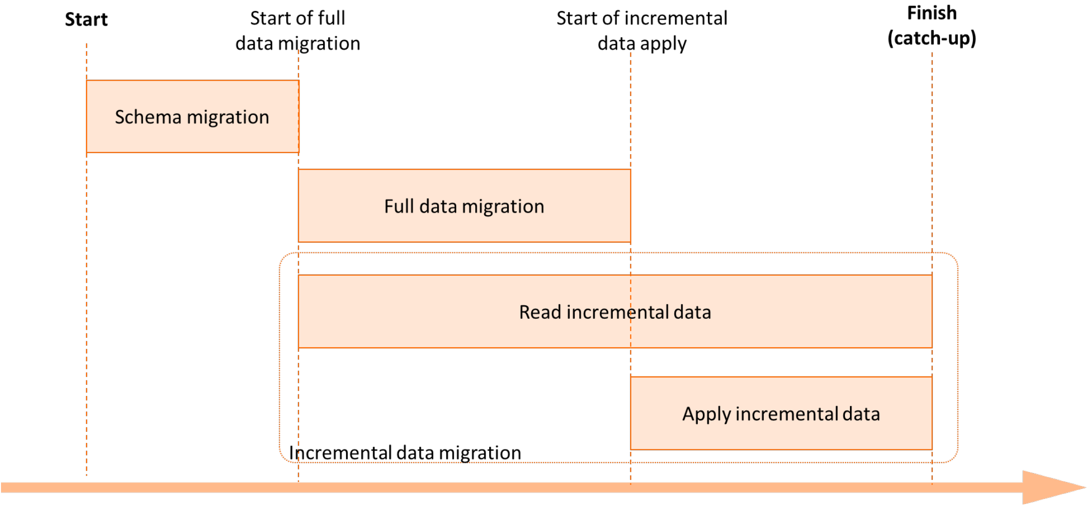
In the full data migration phase, DTS replicates the existing data from the source database to the target database. The source database remains operational and updates are continuously made during the migration process. DTS uses an incremental data reader to capture the ongoing changes that occur during the full data migration process. The incremental data reading is activated when the full data migration starts. During the full data migration phase, incremental data is parsed, reformatted, and stored locally on the DTS server.

Incremental data migration

When the full data migration process is complete, DTS retrieves the incremental data stored locally, reformats it again, and applies the incremental changes in the target database. This process continues until all ongoing updates are replicated to the target database and the source and target databases are in sync.

The data migration mode is illustrated in the following figure:

Data migration process



How DTS works in data synchronization mode

The data synchronization mode of DTS replicates ongoing changes between two data stores. This mode is typically used for OLTP-to-OLAP replications. In this mode, a migration task consists of the following two phases:

- Initial data load: DTS loads the existing data from the source database to the target database.
- Ongoing replication: DTS replicates ongoing changes and keeps the source and target databases in sync.

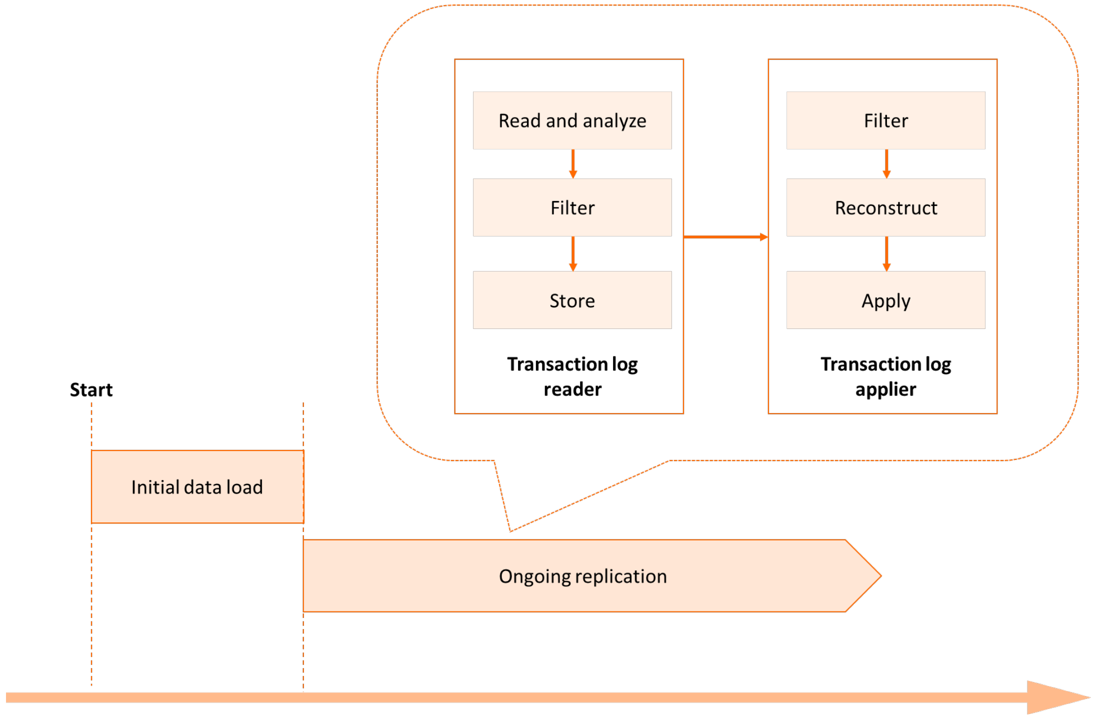
To replicate ongoing changes, DTS uses two components that work with the transaction log:

- Transaction log reader: The transaction log reader communicates with the source database using the corresponding protocol to read the transaction log. For example, it uses Binglog Dump to read transaction log data from ApsaraDB for MySQL databases.
- Transaction log applier: The transaction log applier retrieves data updates from the transaction log

reader, filters the updates to keep only ones related to the objects being replicated, and applies the updates to the target database. When doing this, the transaction log applier maintains the ACID properties of transactions. Both the transaction log reader and the transaction log applier are based on redundant deployments. The HA manager checks the health condition of each server. If an anomaly occurs, the execution of the transaction log is resumed on a healthy server.

The data synchronization mode is illustrated in the following figure:

Data synchronization process



How DTS works in change tracking mode

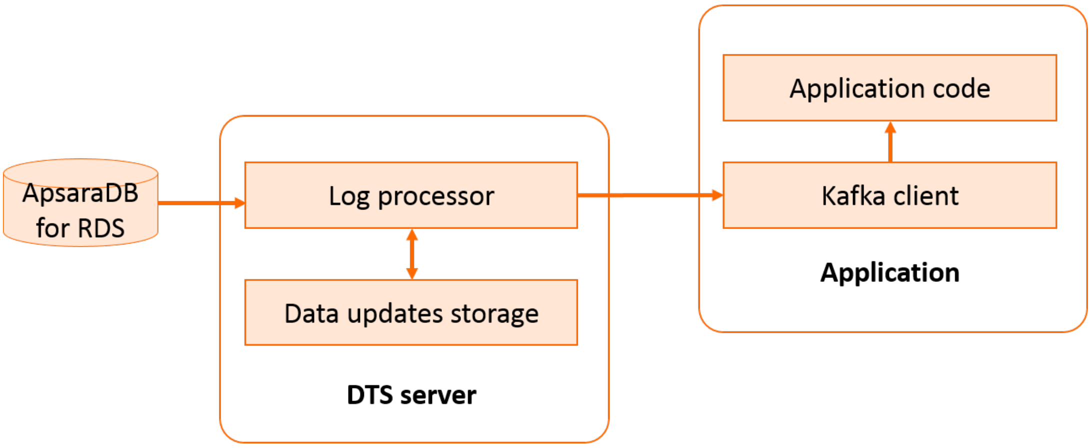
The change tracking replication mode of DTS captures data updates and exposes them as a publisher/subscriber stream. You can customize the consumption mechanism for your different applications.

The log processor communicates with the source database using the corresponding protocol to read the transaction log. For example, it uses Binglog Dump to read transaction log data from ApsaraDB for MySQL databases. Then, the log processor parses the transaction log data, filters the data, normalizes the update records, and keeps the processed data in persistence.

The log processor is based on redundant deployments. The HA manager checks the health condition of each server. If an anomaly occurs, the workloads of transaction log reading are resumed on a healthy server.

The change tracking mode is illustrated in the following figure:

Change tracking process


ApsaraDB for RDS
Log processor
Data updates storage
DTS server
Application code
Kafka client
Application

4.Benefits

Alibaba Cloud Data Transmission Service (DTS) helps you migrate data between data stores, such as relational databases, NoSQL databases, and data warehouses. You can use DTS to migrate your data to Alibaba Cloud or between combinations of cloud and on-premises data systems.

As a managed service, DTS offers several advantages compared with a traditional data migration tool. DTS runs with better performance and guaranteed availability and works seamlessly with Alibaba Cloud database services.

Compatibility with diverse database engines

DTS supports heterogeneous migrations, such as Oracle to MySQL and Oracle to PolarDB O Edition. To reconcile the differences between database engines, DTS provides schema conversion. For example, you can convert an Oracle synonym to a PolarDB O Edition synonym.

Support for multiple use cases

DTS supports several data replication modes, including data migration, data synchronization, and change tracking. You can choose a combination of data replication modes that best suit your use cases.

You can use the data synchronization mode to enable ongoing replication between two data stores and keep them in sync. The replication can work either one-way or two-way.

> ? **Note** Two-way synchronization is only supported for synchronization between two MySQL databases, two PolarDB for MySQL databases, and two ApsaraDB for Redis Enhanced Edition (Tair) databases.

The data synchronization mode allows you to build a distributed system with nodes that replicate to each other in real time, so that you implement high availability, load balancing, and real-time data warehousing based on this setup.

Minimized-downtime migrations

The data migration mode can migrate your data with minimized downtimes. Your source database can remain operational during migration. The actual downtime during data migration is reduced to minutes.

Superior performance

DTS provisions high-performance servers to support data replication workloads. In addition, DTS has introduced a myriad of optimizations to the core infrastructure so that it can deliver a peak data transmission rate of 70 MB per second.

The data synchronization mode can replicate ongoing updates with ultra-low latency. The replication works at transaction level so that even updates to the same table can be broken down to transactions and processed in parallel. This technique significantly improves the concurrency of replication workloads and helps DTS deliver a peak rate of 30,000 records per second (RPS).

DTS supports multiple concurrency and compressed data transmission to reduce bandwidth usage.

> ? **Note** The performance measurements listed here are only benchmarking references. The actual data replication performance depends on multiple factors, such as the performance of the source and target databases, network conditions, and your provisioned instance size.

High availability

DTS runs on high-availability deployments, where each cluster has multiple server nodes. If one node fails, the HA manager switches the workloads to a healthy node in the cluster within seconds.

For certain critical replication links, DTS continuously performs validation to check for data integrity and makes corrections automatically.

The data flows between different functions of DTS are secured and resumable.

Ease of use

The Data Transmission Service console provides an intuitive wizard for you to create replication tasks with a few clicks.

You can easily monitor your replication tasks and view key statistics, such as replication status, progress, and performance.

DTS continuously monitors replication task status. If DTS detects a network or system failure, it automatically fixes the issue, restarts the task, and resumes from where it left off. If the issue persists, you can manually repair and restart the task in the DTS console.

5.Use cases

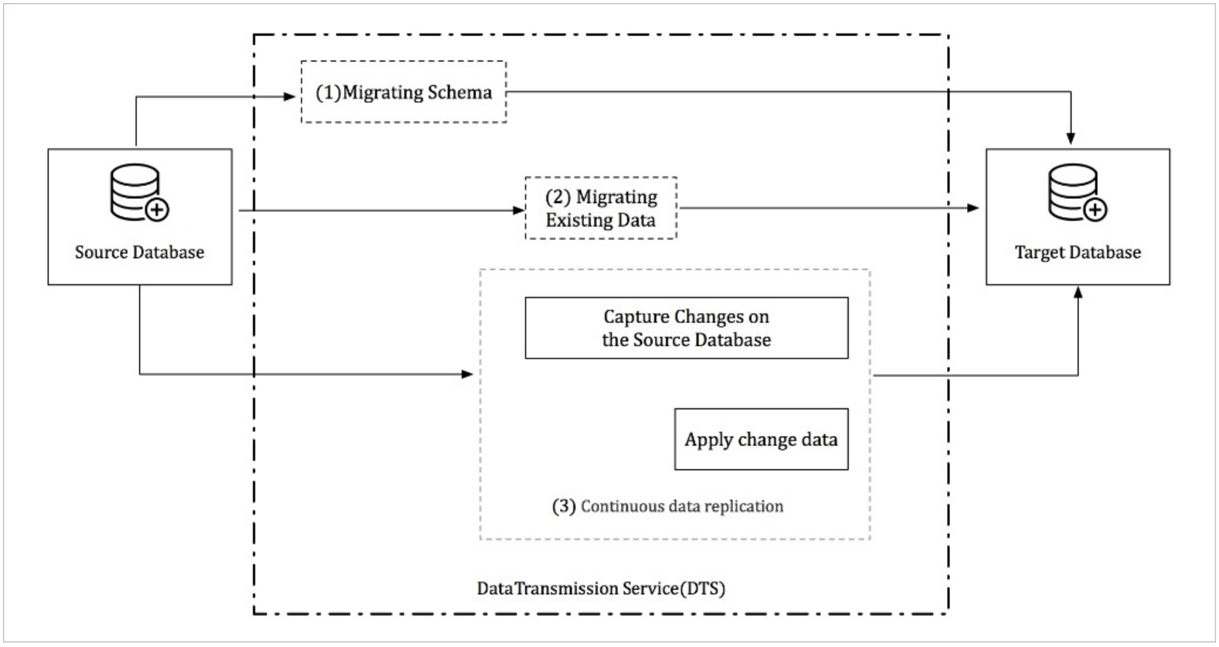
Data Transmission Service (DTS) supports data replication in several modes, including data migration, change tracking, and data synchronization. You can use a combination of data replication modes that best suit your use cases.

Database migration with minimized downtime

Applicable replication mode: data migration

To ensure data consistency, traditional migration processes require that you stop writing data to the source database during data migration. Depending on the data volume and network conditions, the migration process may take several hours or even days. This prolonged process may have a great impact on your business.

DTS can migrate your data with minimized downtime. Your applications remain operational during migration. The only downtime is when you switch your application to the new target database. You can usually limit this switchover window to minutes. The following figure shows the process of data migration.


(1)Migrating Schema
Source Database
(2) Migrating Existing Data
Target Database
Capture Changes on the Source Database
Apply change data
(3) Continuous data replication
DataTransmission Service(DTS)

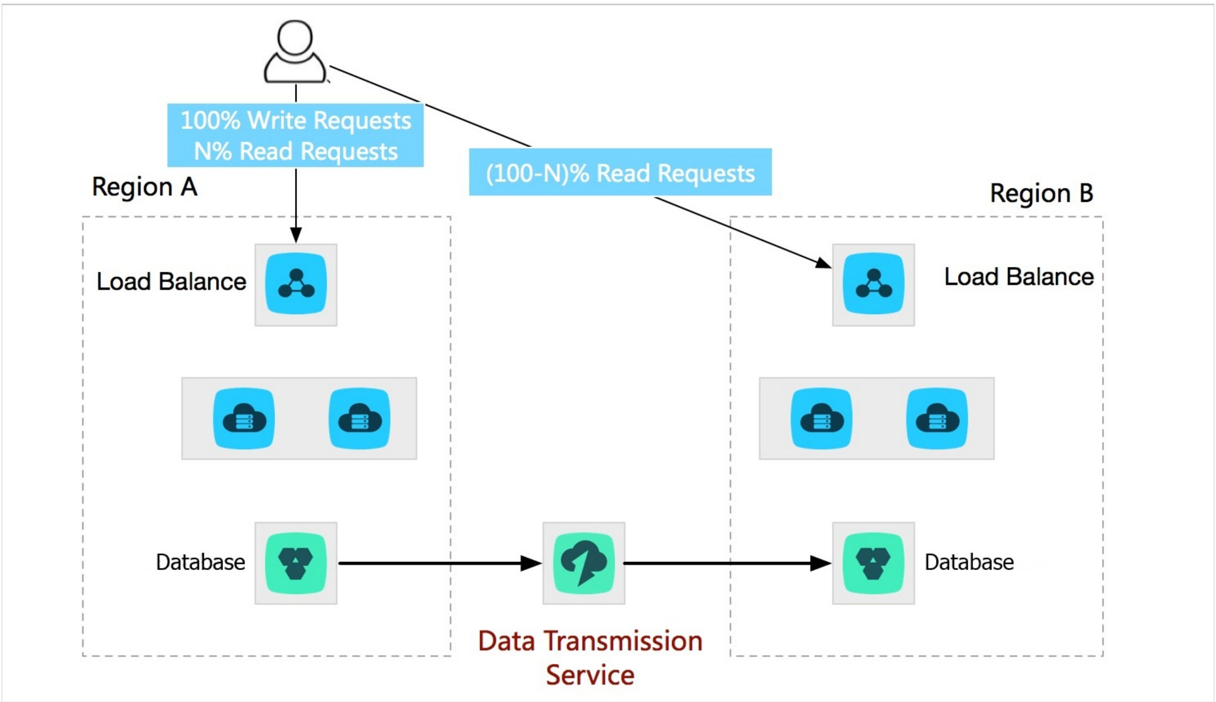
The data migration process includes several phases, namely schema migration, full data migration, and incremental data migration. During incremental data migration, ongoing changes that occur in the source are replicated to the target in near real time. When the migration is complete, you can verify that the migrated data schema and data in the target database are fully compatible with your application. Then, you can switch your application to the target database.

Geo-redundancy

Applicable replication mode: data synchronization

If your application is deployed in a single region, service interruption may occur because of failures that are rare but inevitable, such as power failures and network disconnections.

In this case, you can build a secondary deployment in a different region to increase the availability of your application. DTS continuously replicates changes between these two geo-redundant deployments and keep the regional replicas in sync. If a failure occurs in the primary region, you can switch user requests to the secondary region.



Active geo-redundancy

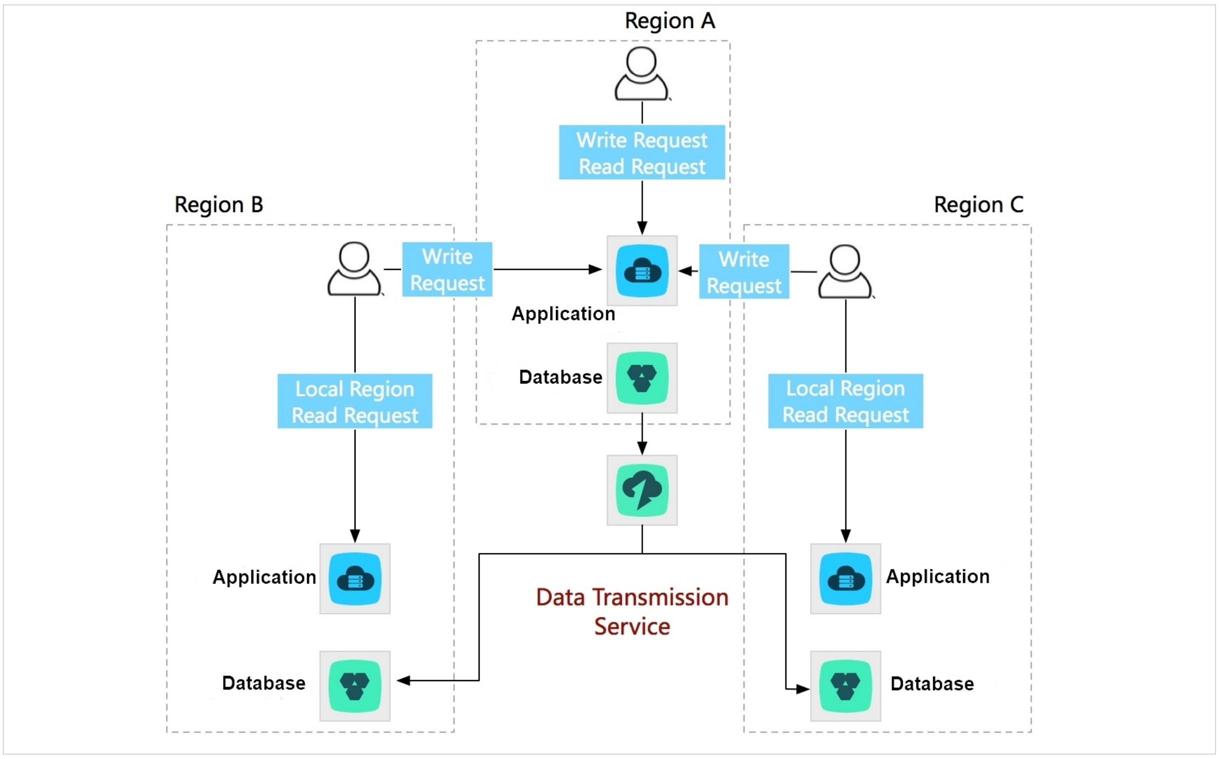
Applicable replication mode: data synchronization

As your business grows, you may encounter the following issues if you deploy your application in a single region:

- Users access your application from a wide range of geographical locations. Distant users may experience high network latency and slow performance.
- The scalability is limited by the capacity of infrastructure in a single region, such as network bandwidth.

To address these challenges, you can build multiple business units in the same city or across different cities. DTS performs two-way real-time data replication between business units to keep the regional replicas in sync. If a failure occurs in a business unit, you only need to switch the traffic to another business unit. The application can be recovered within seconds. In this way, you achieve higher availability based on redundancy of multiple business units.

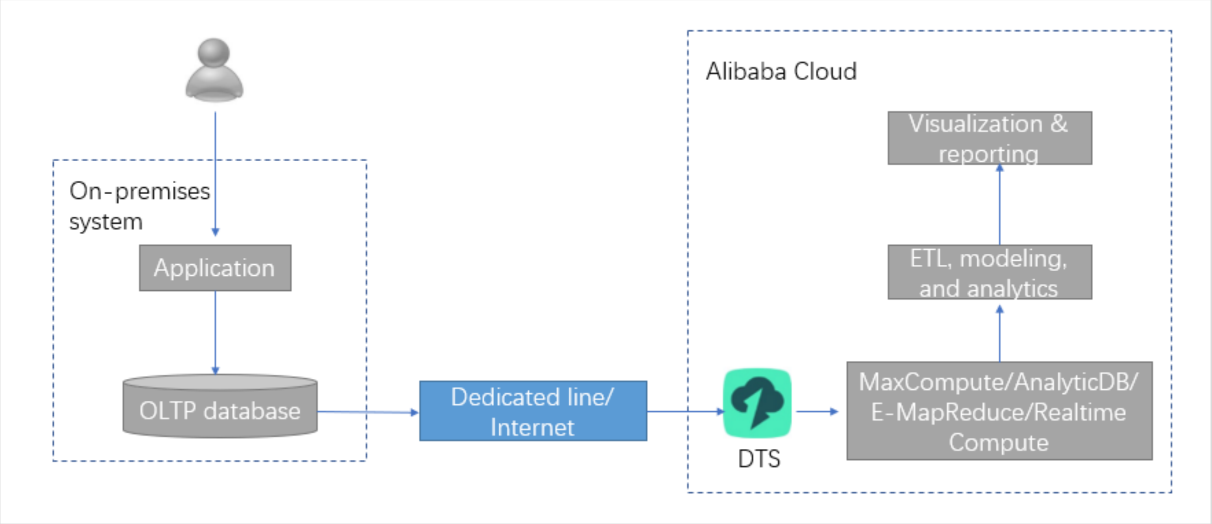
You can also distribute traffic across business units. For example, you can redirect the traffic bound for each business unit to the server closest to the user. This reduces network latency and improves user experience.



Integration with cloud BI systems

Applicable replication mode: data synchronization

Alibaba Cloud provides a complete portfolio of BI offerings. To leverage cloud-based BI capabilities, you need to integrate your applications and the BI data stores of Alibaba Cloud. DTS allows you to replicate data in real time from user-created databases to Alibaba Cloud BI data stores, such as MaxCompute.

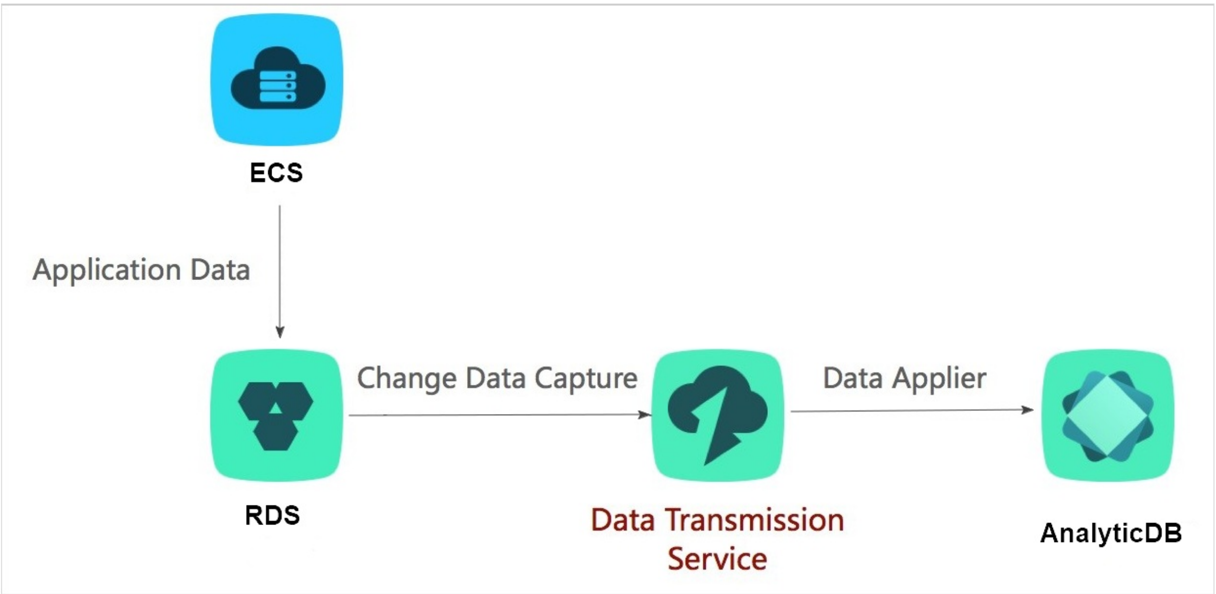

Alibaba Cloud
On-premises system
Application
OLTP database
Dedicated line/ Internet
DTS
Visualization & reporting
ETL, modeling, and analytics
MaxCompute/AnalyticDB/ E-MapReduce/Realtime Compute

Real-time analytics

Applicable replication mode: change tracking

Data analytics is essential in improving enterprise insights and user experience. With the ability to analyze real-time data, enterprises can adjust marketing strategies and adapt to changing markets and higher demands for better user experience.

With the change tracking mode of DTS, you can subscribe to incremental data without disrupting your online transactions. You can use a standard Kafka client SDK to stream the subscribed incremental data to the analytics system so you can perform analytics tasks on the latest transactional data.

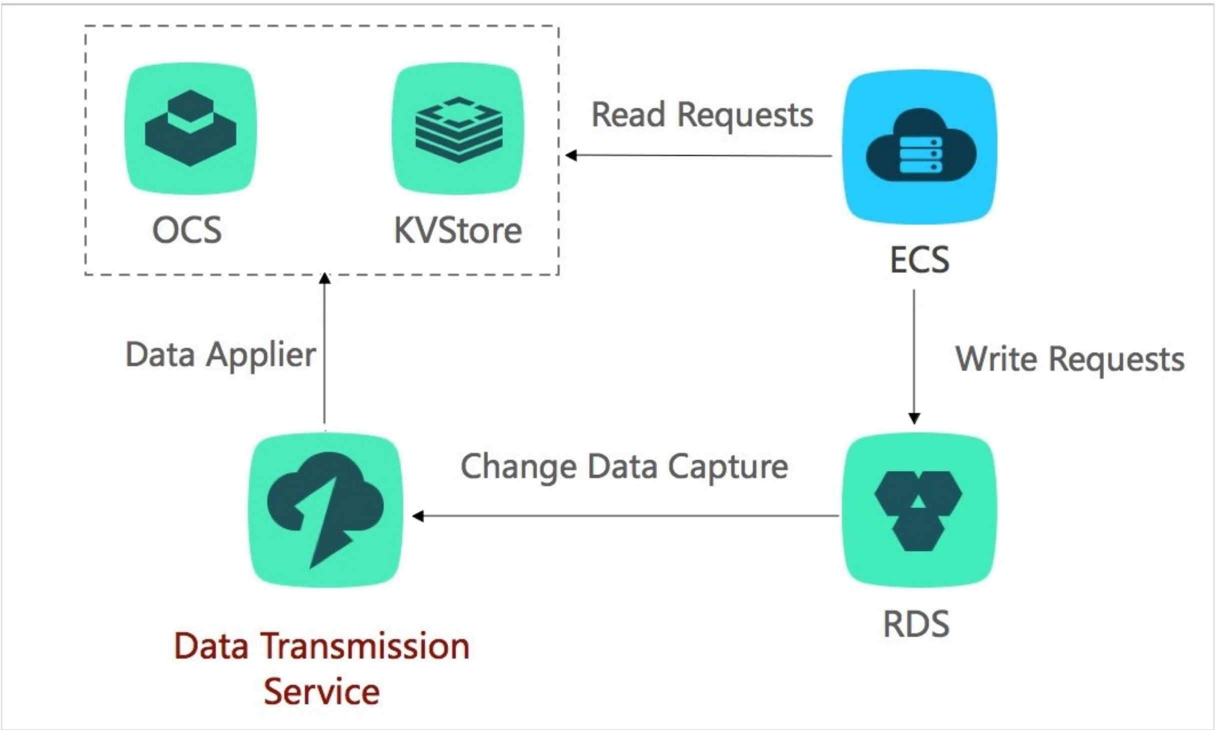


Simple cache updating

Applicable replication mode: change tracking

To improve the response times of your application, a commonly used strategy is to introduce a memory-cached read handler to increase the performance of concurrent read requests. All read requests are directed to the read handler. The read handler can process read requests with better performance using its in-memory retrieval.

With the change tracking mode of DTS, you can implement a simple cache updating mechanism by subscribing to the changes committed to the primary database and updating the cache in near real time.



This architecture provides the following benefits:

- Low-latency write commits

  The write request is completed immediately after the update is committed to the database. The cache updating operation is handled in a parallel process. The application does not have to wait for the cache to be updated.

- Simplified application logic

  The application does not have to write the same update into two replicas. Instead, it only needs to subscribe to the stream of updates and update the cache accordingly in a parallel process.

- No performance overheads on database

  DTS retrieves data updates from the redo log. This process does not interfere with the normal transactions of the read/write database and does not incur performance overheads.

Decoupled components and asynchronous processing

Applicable replication mode: change tracking

A typical online shopping system consists of such components as order management, inventory management, and shipping management. In a synchronous processing model, an order placement action has to integrate all dependent actions that are to be performed in the corresponding components. This means that the user has to wait until all the updates are committed before they have the order confirmed. The synchronous processing model has the following weaknesses:

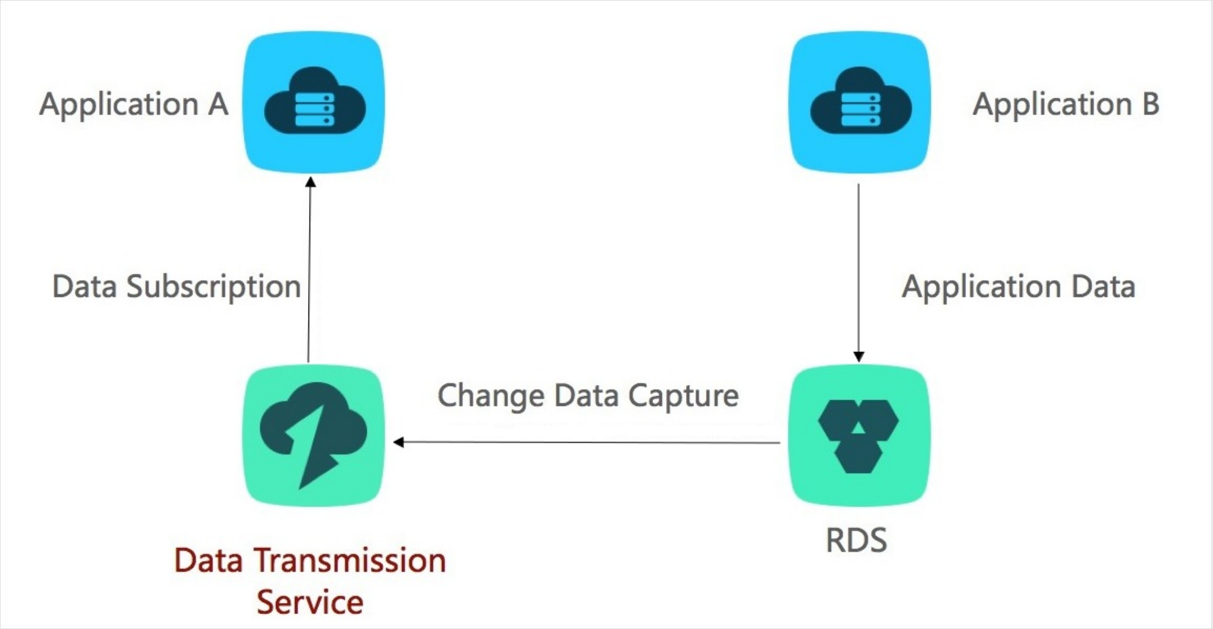
- The order placement action takes a long time to complete.
- With multiple actions integrated into one step, a single fault in any of the components may fail the entire function.

To resolve these issues, you can deploy a system with decoupled components, where each transaction triggers asynchronous updates to the corresponding component databases. Using the change tracking mode of DTS, components can subscribe to data updates and perform follow-up processing accordingly. This architecture decouples core components from other components so that core components can function more stably.

The following figure shows the asynchronous processing across decoupled components.

In this scenario, the order management component returns the result immediately after the user places an order. The change tracking replication mode of DTS captures data updates and exposes them as a publisher/subscriber stream. Other components can use a standard Kafka client to subscribe to the data updates and then perform follow-up processing, such as inventory updates and shipping processing.

This architecture has been adopted in a wide range of businesses of Alibaba Group. Every day, tens of thousands of applications subscribe to the updates made to the Taobao order management system using the change tracking mode of DTS and perform follow-up processing in parallel.



Scalable read capacity

Applicable replication mode: data synchronization

To handle massive amounts of concurrent read requests, you can distribute the workloads among multiple read-only database instances. To achieve this, you can use the data synchronization mode of DTS to replicate data to read-only instances in real time. This scale-out architecture allows you to handle ultra-high concurrent read workloads.

Scheduled replication for data warehousing

Applicable replication mode: data migration

For a large online application that handles massive amounts of transactional data every day, you may need to adopt a next-day warehousing strategy to transfer data into data warehouses on a regular basis. For example, you want to schedule migrations to start during off-peak hours to replicate transactional data of the day to data warehouses. With this deployment, your analytics systems can work on one-day-old data.

6.Limits

This topic describes the limits of Data Transmission Service (DTS).

Data migration mode

The following sections list the limits relating to the data migration mode of DTS.

Source: SQL Server

Limits	Negative effect of failing to follow the limits
The following data types are not supported: sql_variant, hierarchyid, geometry, cursor, and rowversion.	• Changes to values of the image, text, ntext, and xml data types are not written into the log. DTS can only ensure the values of these data types are consistent between the source and target databases. • If unsupported data types are used in the source database, this part of the data is not migrated.
Heap tables are not supported.	These tables cannot be migrated..
Tables that contain non-clustered indexes are not supported.	These tables cannot be migrated.

Source: Oracle

Limits	Negative effect of failing to follow the limits
The following data types are not supported: URI, UDT, BFile, and UROWID.	Data fails to be migrated.

Source: Db2

Limits	Negative effect of failing to follow the limits
The AIX all-in-one edition is not supported.	
Only the following data types are supported: • SMALLINT, INTEGER, BITINT, DECIMAL, CHAR, VARCHAR, DECFLOAT, and DOUBLE REAL • GRAPHIC, VARGRAPHIC (Data records of the GRAPHIC and VARGRAPHIC types are stored as double-byte strings. The boundary check and validity check are not performed.), and DECFLOAT (The data range is large. Some databases do not support this data type. Only numeric values can be truncated.)	Data fails to be migrated.

Limits	Negative effect of failing to follow the limits
The primary/secondary switchover logic supports only the SYNC, NEARSYNC, and ASYNC synchronization modes. Note In the ASYNC mode, if a forced switchover is performed when the primary and secondary databases are not synchronized, data loss may occur on the secondary database. A non-forced switchover is performed after the primary and secondary databases are synchronized.	If SUPERASYNC is used for synchronization, DTS cannot ensure the continuity of the program during switchover.

Source: MongoDB

Limits	Negative effect of failing to follow the limits
You cannot migrate data from MongoDB databases in a cluster architecture. You must migrate each shard of a sharded cluster instance.	N/A
Transaction information is not retained.	When transactions are migrated to the destination database, they are converted into a single record.

Target: MySQL

Limits	Negative effect of failing to follow the limits
During incremental data migration, only the following DDL operations can be synchronized: • ALTER TABLE and ALTER VIEW • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW • DROP INDEX and DROP TABLE • RENAME TABLE • TRUNCATE TABLE	If you perform unsupported DDL operations on the source database, data fails to be migrated.

Target: Distributed Relational Database Service (DRDS)

Limits	Negative effect of failing to follow the limits
The database in DRDS instance are created based on ApsaraDB RDS for MySQL instances (MySQL 5.x) that you purchased. DTS does not support databases that are created based on ApsaraDB RDS for MySQL instances (MySQL 8.0) or ApsaraDB PolarDB for MySQL clusters.	The data migration task cannot be configured.
DDL operations cannot be synchronized.	Data fails to be migrated.

Target: SQL Server

Limits	Negative effect of failing to follow the limits
During incremental data migration, only the following DDL operations can be synchronized: - CREATE TABLE (CREATE TABLE operations for creating partition tables or tables that contain functions cannot be synchronized.) - ALTER TABLE **Note** ALTER TABLE operations only include ADD COLUMN, DROP COLUMN, and RENAME COLUMN. - CREATE INDEX, DROP TABLE, RENAME TABLE, and TRUNCATE TABLE	Data fails to be migrated.

Target: Oracle

Limits	Negative effect of failing to follow the limits
During incremental data migration, only the following DDL operations can be synchronized: - CREATE TABLE (CREATE TABLE operations for creating partition tables or tables that contain functions cannot be synchronized.) - ALTER TABLE **Note** ALTER TABLE operations only include ADD COLUMN, DROP COLUMN, RENAME COLUMN, and ADD INDEX. - CREATE INDEX, DROP TABLE, RENAME TABLE, and TRUNCATE TABLE	Data fails to be migrated.

Target: PostgreSQL

Limits	Negative effect of failing to follow the limits
DDL operations cannot be synchronized during incremental data migration.	Data fails to be migrated.

Target: Db2

Limits	Negative effect of failing to follow the limits
DDL operations cannot be synchronized during incremental data migration.	Data fails to be migrated.

Target: Redis

Limits	Negative effect of failing to follow the limits
During full data migration, the destination Redis database (such as Redis Community Edition and ApsaraDB for Redis Community Edition) cannot contain data.	The collection class or data in the source and destination databases is inconsistent.

Target: MongoDB

Limits	Negative effect of failing to follow the limits
The writing of transactions is not supported.	DTS only ensures eventual consistency between the source and destination databases.

Target: AnalyticDB for MySQL

Limits	Negative effect of failing to follow the limits
A maximum of 1,024 tables can be migrated.	Data fails to be migrated.
AnalyticDB for MySQL is incompatible with MySQL in terms of the processing of some invalid values.	Data fails to be migrated.
The maximum length of a column is 16 MB by default.	Data fails to be migrated.

Data synchronization mode

The following sections list the limits relating to the data synchronization mode of DTS.

Target: MySQL

Limits	Negative effect of failing to follow the limits
Only the following DDL operations can be synchronized: • ALTER TABLE and ALTER VIEW • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW • DROP INDEX and DROP TABLE • RENAME TABLE • TRUNCATE TABL	If you perform unsupported DDL operations on the source database, data fails to be synchronized.

Target: Redis

Limits	Negative effect of failing to follow the limits
If the Redis database is deployed in the cluster architecture, you cannot run the FLUSHDB, FLUSHALL, or SWAPDB command during data synchronization.	Data in the source and destination databases is inconsistent.

Limits	Negative effect of failing to follow the limits
During initial full data synchronization, the destination Redis database (such as Redis Community Edition and ApsaraDB for Redis Community Edition) cannot contain data.	The collection class or data in the source and destination databases is inconsistent.

Target: AnalyticDB for MySQL

Limits	Negative effect of failing to follow the limits
A maximum of 1,024 tables can be synchronized.	Data fails to be synchronized.
Only the following DDL operations can be synchronized: CREATE TABLE, DROP TABLE, RENAME TABLE, TRUNCATE TABLE, ADD COLUMN, and DROP COLUMN.	Data fails to be synchronized.
Each table to be synchronized must contain a primary key. The value of the primary key cannot be NULL.	Data fails to be synchronized or the data synchronization task fails to be started.
AnalyticDB for MySQL is incompatible with MySQL in terms of the processing of some invalid values.	Data fails to be synchronized.
The maximum length of a column is 16 MB by default.	Data fails to be synchronized.

Target: MaxCompute

Limits	Negative effect of failing to follow the limits
Only the following DDL operations can be synchronized: ALTER TABLE and ADD COLUMN.	Data fails to be synchronized.

7.Terms

The following table describes the terms that are frequently referred to in the Data Transmission Service (DTS) documentation.

Term	Description
consumption checkpoint	The consumption checkpoint is the timestamp of the latest data update that is consumed by a consumer application. Every time that a consumer client consumes and commits a data update, DTS marks the latest commit time as the consumption checkpoint. If the consumer client stops responding, DTS automatically resumes from the consumption checkpoint when sending updates to the next healthy consumer.
data update	Data updates are data manipulation language (DML) operations that modify data, without modifying the schema, such as INSERT, DELETE, and UPDATE operations.
full data migration	In the full data migration phase, DTS replicates the existing data from the source database to the target database. If you select only schema migration and full data migration for the task, updates that occur during the full data migration phase will not be captured and replicated to the target database. **Note** To ensure data consistency, we recommend that you suspend all updates to the source database during the full data migration phase. To migrate data with minimized downtimes, you must select schema migration, full data migration, and incremental data migration when creating the data migration task.
incremental data migration	In the incremental data migration phase, DTS continuously replicates ongoing changes between the source and target data stores. This function is typically used in minimized-downtime migrations to replicate changes that occur during the full data migration phase. **Note** Data synchronization is a continuous process that does not automatically end. To stop the process, you need to manually stop the data migration task.
initial synchronization	DTS replicates the existing schema objects and data to the target database before replicating ongoing changes. Initial synchronization includes initial schema synchronization and initial full data synchronization. • Initial schema synchronization: replicates the schema objects corresponding to your selected objects from the source database to the target database. • Initial full data synchronization: replicates the existing data of your selected objects from the source database to the target database.

Term	Description
precheck	The system performs a precheck before starting a data migration task, data synchronization task, or change tracking task. In this process, the following items are checked: • Connectivity between the DTS server and the source and target databases • Database account permissions • Whether binary logging is enabled • Database version numbers **Note** If the precheck fails, click the information icon ⓘ next to each failed item to view the details. Follow the instructions to fix the issues and run the precheck again
replication mode	A task can be created in one of the replication modes, each of which is suited to different business scenarios. Migration modes include data migration mode, data integration mode, data synchronization mode, and change tracking mode.
replication performance	The replication performance is a measurement based on the number of records that are replicated to the target database per second. The unit is records per second (RPS). For more information, see Specifications of data synchronization instances.
schema migration	In the schema migration phase, DTS migrates schema objects from the source database to the target database, including tables, views, triggers, and stored procedures. To migrate schema objects across heterogeneous databases, DTS converts schema objects into data types that work with the target database. For example, it converts the NUMBER data type in Oracle databases to the DECIMAL data type in MySQL databases.
schema update	Schema updates are data definition language (DDL) operations that modify schema objects, such as CREATE TABLE, ALTER TABLE, and DROP VIEW operations.
synchronization latency	The synchronization latency is a measurement based on the difference between the timestamp of the latest data record synchronized to the target database and the current timestamp of the source database. If the synchronization latency is zero, the source database is in sync with the target database.
timestamp range	The timestamp range is the range of timestamps for the incremental data that is stored in a change tracking task. By default, the change tracking task keeps updates for 24 hours. DTS regularly cleans expired incremental data and updates the timestamp range of the change tracking channel. **Note** The timestamp of the incremental data is the timestamp when the incremental data is updated in the source database and written into the transaction log.

8.FAQ

This topic provides answers to frequently asked questions (FAQ) about Data Transmission Service (DTS).

- General
  - Which databases do DTS tasks support in data migration, data synchronization, and change tracking mode?
  - What are the differences between incremental data migration and data synchronization?
  - How is the synchronization delay measured?
  - How do I consume tracked data by using a task in change tracking mode?
- Billing
  - How is DTS billed?
  - Why is the price of data synchronization higher than that of data migration?
- Instance sizes
  - What are the differences in performance between different instance sizes?
  - Can instance sizes be downgraded and upgraded?
- Features
  - Can I migrate or synchronize data between Alibaba Cloud accounts?
  - Can I migrate data within a single instance?
  - Can I migrate or synchronize DML and DDL operations?
  - Can I migrate or synchronize database shards and table shards?
  - Can I migrate or synchronize data across time zones and character sets?
  - Can I change the name of objects that are migrated or synchronized to the target database?
  - Can I change the name of objects that are migrated or synchronized to the target database?
  - Can I filter columns and rows?
  - Can I add or remove objects to be synchronized?
- Performance
  - How do I view the performance metrics of data migration and synchronization tasks?

Which databases do DTS tasks support in data migration, data synchronization, and change tracking mode?

DTS supports data transmission between various data sources, such as relational database management systems (RDBMS), NoSQL databases, and online analytical processing (OLAP) databases. For more information, see Supported sources and targets.

> ⓘ **Note** DTS also supports data migration and synchronization between databases that are provided by third-party cloud vendors. For more information, see Overview of data migration scenarios.

What are the differences between incremental data migration and data synchronization?

The data synchronization mode and the incremental data migration phase of the data migration mode offer seemingly similar functionality. They both perform continuous replications. However, they have different focuses.

The incremental data migration phase of the data migration mode is intended for minimized-downtime migrations. While DTS loads the existing data of the source database, updates are still being made to the source database. The incremental data migration is to ensure the updates occurring during this period are eventually applied to the target database.

The data synchronization mode is intended for continuous replications. It is usually used for data replication between databases in a distributed system, for example, for redundancy and high availability. Compared with the incremental data migration phase of data migration, the data synchronization mode delivers higher performance and lower lag.

How is the synchronization delay measured?

The synchronization delay is the difference between the timestamp of the latest synchronized data in the target database and the current timestamp in the source database. The unit is milliseconds.

How do I consume tracked data by using a task in change tracking mode?

You can use a Kafka client to consume tracked data. For more information, see Use a Kafka client to consume tracked data

How is DTS billed?

DTS provides two billing methods: subscription and pay-as-you-go. Change tracking and data synchronization instances support both billing methods. Data migration mode instances only support pay-as-you-go.

The subscription billing method allows you to pay for your subscription when creating an instance. Subscription lengths are monthly or yearly, and you can renew the subscription either manually or automatically. We recommend that you select subscription billing if you plan to use DTS for a period lasting a month or longer.

The pay-as-you-go billing method allows you to pay for your DTS instance based on the time your tasks are running. A pay-as-you-go instance is billed on an hourly basis, and you can release a pay-as-you-go instance at any time. We recommend that you select this billing method for short term use.

For more information about billing methods, see Billing methods. For details about how DTS is priced, see the Pricing page.

Why is the price of data synchronization higher than that of data migration?

Data synchronization mode comes with more advanced features. For example, you can modify the objects to be synchronized. You can configure two-way data synchronization between MySQL databases. In addition, the data synchronization mode ensures low network latency through data transmission over the internal network.

What are the differences in performance between different instance sizes?

The maximum performance of data migration and data synchronization tasks depends on the instance size.

For information about how different instance sizes perform in different scenarios, see Data migration mode test data and Data synchronization mode test data.

Can instance sizes be downgraded and upgraded?

You can upgrade the instance size of instances in data migration or data synchronization mode. You can upgrade an instance size by finding it in the instance list, and then clicking **Upgrade** in the **Actions** column. Instance sizes cannot be downgraded.

Can I migrate or synchronize data between Alibaba Cloud accounts?

- Data migration mode: You can migrate data across Alibaba Cloud accounts between ApsaraDB RDS for MySQL instances. For more information, see Migrate data between RDS instances of different Alibaba Cloud accounts. You can also migrate data across Alibaba Cloud accounts between other database instances, such as Apsara PolarDB for MySQL, DRDS, Redis, and MongoDB. To do this, you can specify the database as a User-created Database with a Public IP Address when you configure the data migration task.
- Data synchronization mode: You can synchronize data across Alibaba Cloud accounts only between ApsaraDB RDS for MySQL instances. For more information, see Synchronize data between ApsaraDB RDS for MySQL instances that belong to different Alibaba Cloud accounts.

Can I migrate data within a single instance?

Yes, you can migrate data within a single instance. For more information, see Migrate data between databases that have different names.

Can I migrate or synchronize DML and DDL operations?

Yes, you can migrate or synchronize DML and DDL operations between relational databases. The supported DML operations are INSERT, UPDATE, and DELETE. The supported DDL operations are CREATE, DROP, ALTER, RENAME, and TRUNCATE.

> Note The supported DML and DDL operations are different in different scenarios. For example, if you synchronize data from a MySQL database to AnalyticDB for MySQL 2.0, only the following DDL operations are supported: CREATE TABLE, ALTER TABLE, and DROP TABLE. Only the following DML operations are supported: INSERT, UPDATE, and DELETE.

Can I migrate or synchronize database shards and table shards?

Yes, you can migrate or synchronize database shards and table shards. For example, you can migrate or synchronize database shards and table shards from a MySQL database and Apsara PolarDB for MySQL cluster to AnalyticDB for MySQL. This allows you to merge multiple tables.

Can I migrate or synchronize data across time zones and character sets?

Yes, you can migrate or synchronize data across time zones and character sets.

Can I change the name of objects that are migrated or synchronized to the target database?

Yes, you can change the name of columns, tables, and databases by using the object name mapping feature. For more information, see Object name mapping.

Can I filter columns and rows?

Yes, you can filter columns and rows in a table. For more information, see Filter the data to be migrated.

Can I add or remove objects to be synchronized?

Yes, you can add or remove objects to be synchronized. For more information, see Add an object to a data synchronization task and Remove an object from a data synchronization task.

How do I view the performance metrics of data migration and synchronization tasks?

You can view performance metrics of data migration and data synchronization tasks in the DTS console. For information about how to view the metrics, see View the connection status and performance of incremental data migration and View the connection status and performance of data synchronization.