ALIBABA CLOUD

# Alibaba Cloud

## Apsara Stack Agility

Technical Whitepaper

Product Version: 2102, Internal: V3.5.0

Document Version: 20210719

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.IDC requirements

The features and performance of Apsara Stack platforms and services depend on the reliability (24/7 stable operation of servers and network devices) of Apsara stack data centers. This stability relies on the reliability of a series of complex infrastructure such as cooling and power supply. We recommend that you abide to tier 3 or a similar classification when building data centers that host Apsara Stack platforms to reduce stability risks in essence.

# 1.1. Environment requirements

This topic describes the environment requirements for Apsara Stack data centers.

| No. | Description | Requirement | Matching type |
| --- | --- | --- | --- |
| 1 | Areas prone to flooding, such as the downstream of dams or flood-prone regions | Data centers cannot be set up in such areas. | Required |
| 2 | Areas prone to landslides, debris flows, or mountain slopes | Data centers cannot be set up in such areas. | Required |
| 3 | Seismic zones or fault zones | Data centers cannot be set up in such areas. | Required |
| 4 | Distance from areas where have experienced 100-year floods | No less than 100 meters. | Required |
| 5 | Distance from hazardous areas in chemical plants, landfills, gas stations, and polluted sites that have flammables and explosives such as dangerous chemicals and gas. | No less than 400 meters. | Required |
| 6 | Distance from military arsenals | No less than 1,600 meters. | Required |
| 7 | Distance from airports | The distance from both sides of the runway is no less than 1,000 meters. The distance from runways in the direction of takeoff and landing is no less than 8,000 meters. | Required |
| 8 | Distance from public parking lots | No less than 20 meters. | Required |
| 9 | Main roads of the physical park | At least two roads are required. One road must be a two-lane, two-way road, which can accommodate trucks 15 meters long and 3 meters wide. | Recommended |
| 10 | Distance from commercial and residential areas | No greater than 16,000 meters. | Recommended |

| No. | Description | Requirement | Matching type |
| --- | --- | --- | --- |
| 11 | Physical park | The physical park is independent or can be isolated to provide secure isolation. | Recommended |

# 1.2. Building requirements

This topic describes the building requirements for Apsara Stack data centers.

| No. | Description | Requirement | Matching type |
| --- | --- | --- | --- |
| 1 | Gross floor area of a single building | No less than 8,000 square meters. | Recommended |
| 2 | Acceptance of fire protection systems installed in buildings | Fire protection systems installed in buildings are tested and approved by the local fire department. | Required |
| 3 | Floor load capacity | More than 1,000 kg per square meters. | Required |
| 4 | Layer height | The clear span of buildings is greater than 3.6 meters. | Required |
| 5 | Transportation | Freight elevators are required for buildings no less than two floors and have a weight capacity of no less than two tons. The transportation aisles are no less than 2.4 meters wide and no less than 2.5 meters high. | Required |
| 6 | Classification of seismic protection of building constructions | The classification of seismic protection of building constructions is not lower than building type C. | Required |
| 7 | Fire-resistance rating | No less than Level 2. | Required |
| 8 | Waterproof rating | Level 1. | Required |

# 1.3. Power system

This topic describes the requirements for power systems in Apsara Stack data centers.

| No. | Description | Requirement | Matching type |
| --- | --- | --- | --- |
| 1 | Power introduction | At least a written certificate with power supply assurance is required. | Required |

| No. | Description | Requirement | Matching type |
| --- | --- | --- | --- |
| 2 | Route requirements for mains supply introduction | Dual routes are required and their distance must be greater than 10 meters. Cables are routed to the park on different roads. | Required |
| 3 | Requirements for mains supply introduction to substations | Class-A mains supply and two different circuits or two 10 kV, 35 kV, or 110 kV substations are used. | Required |
| 4 | Diesel generators | Diesel generators are configured for N + 1 redundancy. Diesel generators can start under load within two minutes. | Required |
| 5 | Period of time for which oil in tanks can be used | Greater than eight hours. | Required |
| 6 | Uninterruptible power supply (UPS) and redundancy | A UPS system based on 2N redundancy configuration is used for AC distribution. Or a high-voltage direct current (HVDC) system is used for a single mains supply. | Required |
| 7 | Period of time for which storage batteries can be discharged | No less than 15 minutes. | Required |
| 8 | Cabinet power distribution | A dual-circuit power supply system is used, which includes transformers, distribution lines, uninterruptible power supply, rack-mountable power distribution cabinets, and rack power distribution units (PDUs). | Required |
| 9 | Cabinet power consumption | No less than eight kW. | Recommended |

# 1.4. Cooling system

This topic describes the requirements for the cooling system in Apsara Stack data centers.

| No. | Description | Requirement | Matching type |
| --- | --- | --- | --- |
| 1 | Air conditioners, water pumps, water chiller units, and cooling towers in data centers | Air conditioners, water pumps, water chiller units, and cooling towers in data centers are configured for N + 1 redundancy. | Required |
| 2 | Power distribution for precision air conditioners in a chilled water system | Uninterrupted power supply (UPS) | Required |
| 3 | Power distribution for water supply pumps in a water-cooling system | UPS | Required |
| 4 | Period of time for which cool storage equipment can provide cooling | Time period during which cool storage equipment can provide cooling is no less than 10 minutes. When the cooling system is interrupted, the temperature of cold aisles in data centers cannot exceed 30 degrees Celsius. | Required |
| 5 | Building automation system | UPS. Redundancy must be provided for the direct digital controller (DDC) system and servers. | Recommended |

# 1.5. Monitoring requirements

This topic describes the monitoring requirements for Apsara Stack data centers.

| No. | Description | Requirement | Matching type |
| --- | --- | --- | --- |
| 1 | Monitoring access standards | Network communication is enabled based on TCP or IP sockets. | Recommended |
| 2 | Monitoring scope | The following items in data centers are monitored: temperature and humidity inside the data centers, terminal devices of the air conditioning system, chillers, pumps of the air conditioning system, power distribution cabinets, high-voltage direct current (HVDC) systems, uninterrupted power supply (UPS) systems, transformers, diesel generators, and mains supply. | Required |

# 1.6. O&M requirements

This topic describes the O&M requirements for Apsara Stack data centers.

| No. | Description | Requirement | Matching type |
| --- | --- | --- | --- |
| 1 | Technical team | A technical team must consist of the following personnel: one person for building decoration, one to two persons for air conditioning and refrigeration, one to two persons for high voltage power system, and at least one person for low voltage system monitoring. | Recommended |
| 2 | Construction delivery capability | The business deployment requirements are met (1,000 cabinets delivered within six months). | Recommended |
| 3 | Service-level agreement (SLA) | The availability of power, cooling, and network is above 99.99%. | Recommended |
| 4 | O&M personnel in the O&M system | The level and number of O&M personnel are confirmed. | Required |
| 5 | Professional qualifications of O&M personnel in the O&M system | The number of professional and technical personnel is no less than two in each of the following fields: electrical system, heating, ventilation, and air conditioning (HVAC), fire protection, and low voltage system. | Recommended |
| 6 | Duty system of the O&M system | The personnel on duty and emergency response mechanism are available 24/7/365 for infrastructure and network maintenance in data centers. | Required |
| 7 | Hardware and software maintenance of devices in the O&M system | A 24/7/365 professional maintenance service is purchased. | Required |
| 8 | Building management system (BMS) and video surveillance in the O&M system | The power and environment supervision system or BMS is used to monitor the running status of key infrastructure. The 24/7 video surveillance is provided, and the records are retained for 90 days. | Required |

| No. | Description | Requirement | Matching type |
| --- | --- | --- | --- |
| 9 | Entry and exit management of personnel and articles in the O&M system | A clear management process is provided, and records are complete and traceable. | Required |
| 10 | Service qualification | IDC business qualification: An Internet Data Centre Value Added Telecom Service license (IDC VATS) issued by the Chinese government is recommended. | Recommended |
| 11 | Third-party certification | The SSAE 16, ISO 17799, and ISO 9001 audits are passed. SSAE 16 ensures that service providers have sufficient security controls and safeguards in place to protect the security of user data. ISO 17799 ensures that the information security of service providers is less likely to be damaged. ISO 9001 sets out the criteria for a quality management system (QMS) of service providers. | Recommended |
| 12 | Operator personnel handover interface | A clear personnel handover interface is provided, including the assignment of roles and responsibilities, and the problem escalation path to the personnel who is in charge of the project and who holds the highest rank on the operator side. All responsibilities must be assigned and confirmed at the beginning of the project and be carried out for the entire project. | Recommended |

# 1.7. Communication requirements

This topic describes the communication requirements for Apsara Stack data centers.

| No. | Description | Requirement | Matching type |
| --- | --- | --- | --- |
| 1 | Number of direct routes between two data centers | Two direct routes with distances greater than 500 meters. Cables cannot be routed on the same conduit, trench, or route. | Required |

| No. | Description | Requirement | Matching type |
| --- | --- | --- | --- |
| 2 | Number of outgoing routes in a single data center | Provide two outgoing routes. The number of outgoing routes can be expanded to three as required before the project is delivered. | Recommended |
| 3 | Number of outgoing optical fibers | No less than 20 pairs. | Recommended |
| 4 | Routing method of optical cables | Optical cables must be placed into buried conduits. Overhead cabling is not allowed. | Required |
| 5 | Connection method of optical cables | Optical cables must be connected inside data centers. Outdoor connections are not allowed. | Required |
| 6 | Outgoing conduits | The park has more than two outgoing conduits in different directions and their distance is greater than 50 meters. Each outgoing conduit corresponds to a different entrance room. | Recommended |
| 7 | Communication rooms | There are two separate communication rooms in each data center. | Recommended |
| 8 | Leased lines and bandwidth | The data centers have the capabilities to support leased lines, Border Gateway Protocol (BGP) lines, and static bandwidth. | Recommended |
| 9 | Optical cable routes inside data centers | Cables inside data centers must be routed separately to ensure dual routes. The distance between the two routes must be greater than 10 meters. | Required |
| 10 | Access to optical cables of other operators | The data centers can access to optical cables of other operators. | Recommended |

| No. | Description | Requirement | Matching type |
| --- | --- | --- | --- |
| 11 | Number of direct routes between buildings | Two routes are required and four routes are preferred within physical fences. Three routes are required and four routes are preferred outside fences built on property lines. These routes can be completed when data centers are delivered. Direction of the routes must be approved by Alibaba Cloud. | Recommended |
| 12 | Number of optical fibers between buildings | At least 384-core × 4 optical fibers are required and can be scaled out. | Recommended |

# 2.Object Storage Service (OSS)

## 2.1. What is OSS?

Object Storage Service (OSS) is a secure, cost-effective, and highly reliable cloud storage service provided by Alibaba Cloud. It enables you to store a large amount of data in the cloud.

Compared with user-created server storage, OSS has outstanding advantages in reliability, security, cost-effectiveness, and data processing capabilities. OSS enables you to store and retrieve a variety of unstructured data objects, such as text, images, audios, and videos over the network at any time.

OSS is an object storage service based on key-value pairs. Files uploaded to OSS are stored as objects in buckets. You can obtain the content of an object based on the object key.

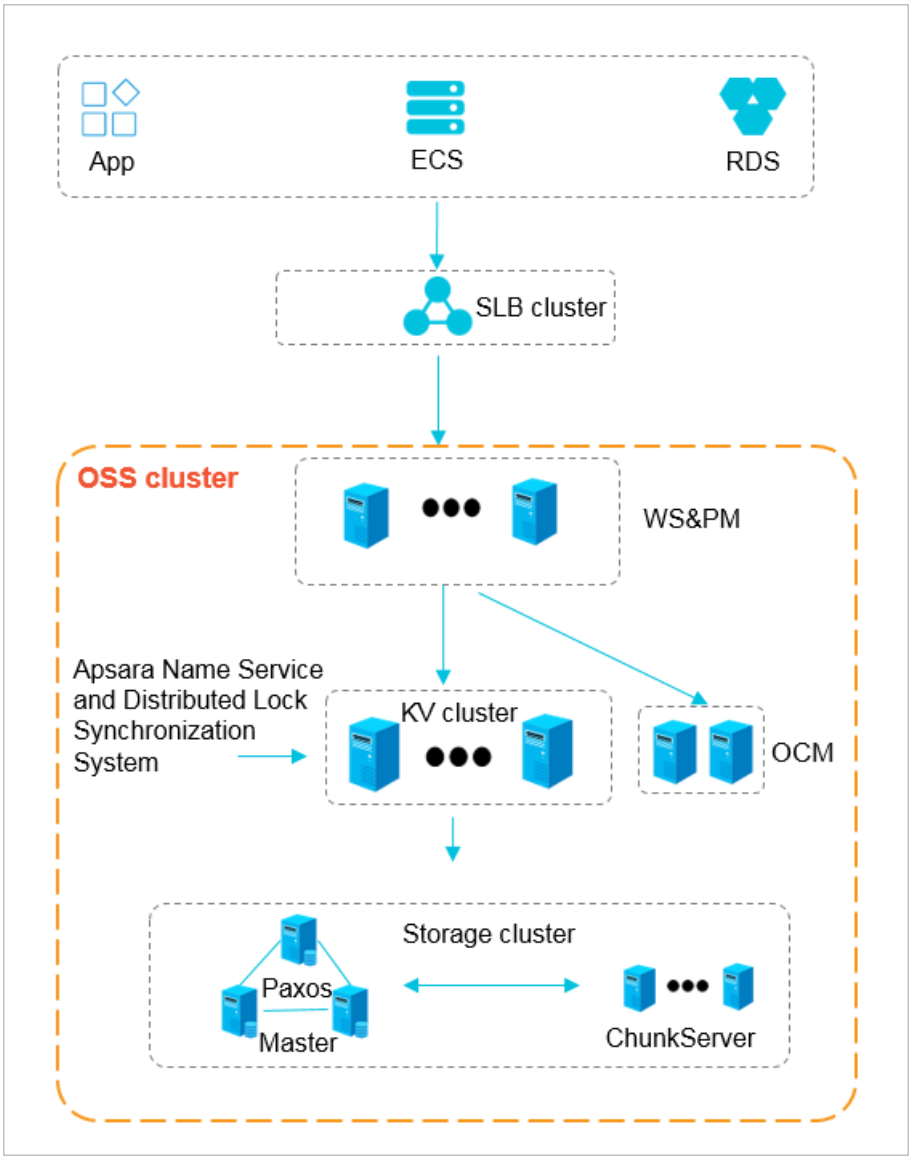
In OSS, you can perform the following operations:

- Create a bucket and upload objects to the bucket.
- Obtain an object URL from OSS to share or download the object.
- Modify the attributes or metadata of a bucket or an object. You can also configure the ACL of the bucket or the object.
- Perform basic and advanced operations in the OSS console.
- Perform basic and advanced operations by using OSS SDKs or calling RESTful API operations in your application.

## 2.2. Architecture

OSS is a storage solution that is built on the Apsara system. It is based on the infrastructure such as Apsara Distributed File System and SchedulerX. The infrastructure provides OSS and other Alibaba Cloud services with importance features such as distributed scheduling, high-speed networks, and distributed storage. The following figure shows the OSS architecture.

OSS architecture




- WS & PM: the protocol layer that receives and authenticates the request sent by using a RESTful protocol. If the authentication is successful, the request is forwarded to KVEngine for further processing. If the authentication fails, an error message is returned.
- KV cluster: used to process structured data, including reading and writing data based on object names. The KV cluster also supports sporadic bursts of requests. When a service has to run on a different physical server due to a change to the service coordination cluster, the KV cluster can coordinate and find the access point.
- Storage cluster: Metadata is stored in the master node. A distributed message consistency protocol of Paxos is adopted between Master nodes to ensure the consistency of metadata. This method ensures efficient distributed storage of and access to objects.

# 2.3. Features

This topic lists the common features of OSS.

Before you start to use OSS, we recommend that you have a good understanding of basic terms used in OSS, such as bucket, object, region, and endpoint. For more information, see Terms.

The following table describes features of OSS.

OSS features

| Category | Feature | Description |
| --- | --- | --- |
| Bucket | Create buckets | Before you upload an object to OSS, you must create a bucket to store the object. |
| | Delete buckets | If you no longer use a bucket, delete it to avoid further fees. |
| | Modify bucket ACL | OSS supports ACL for access control. You can configure the ACL of a bucket when you create it or modify the ACL of a created bucket. |
| | Configure static website hosting | You can configure static website hosting for your bucket and access this static website through the bucket domain name. |
| | Configure hotlink protection | To prevent additional fees caused by unauthorized access to the data in your bucket, you can configure hotlink protection for your buckets based on the Referer field in HTTP requests. |
| | Manage CORS | OSS provides cross-origin resource sharing (CORS) over HTML5 to implement cross-origin access. |
| | Configure lifecycle rules | You can define and manage lifecycle rules for all or a subset of objects in a bucket. You can configure lifecycle rules to manage multiple objects and automatically delete parts. |
| Object | Upload objects | You can upload any type of objects to a bucket. |
| | Create folders | You can manage OSS folders the way you manage folders in Windows. |
| | Search for objects | You can search for objects whose names contain the same prefix in a bucket or folder. |
| | Obtain object URLs | You can obtain the URL of an object to share or download the object. |
| | Delete objects | You can delete a single object or multiple objects. |
| | Delete folders | You can delete a single folder or multiple folders. |
| | Modify object ACL | You can configure the ACL of an object when you upload it or modify the ACL of an uploaded bucket. |
| | Manage parts | You can delete all or some parts from a bucket. |
| Image processing | IMG | You can perform operations such as format conversion, cropping, scaling, rotating, watermarking, style encapsulation on images stored in OSS. |
| Access control for VPC | Single Tunnel | You can create single tunnels to access OSS resources through VPC. |

| Category | Feature | Description |
| --- | --- | --- |
| API | API | OSS supports RESTful API operations and provides examples. |
| SDK | SDK | OSS supports development based on SDKs for various programming languages and provides examples. |

# 3.ApsaraDB for RDS

## 3.1. What is ApsaraDB RDS?

ApsaraDB RDS is a stable, reliable, and scalable online database service. Based on the distributed file system and high-performance storage, ApsaraDB RDS provides a set of solutions for disaster recovery, backup, restoration, monitoring, and migration.

### ApsaraDB RDS for MySQL

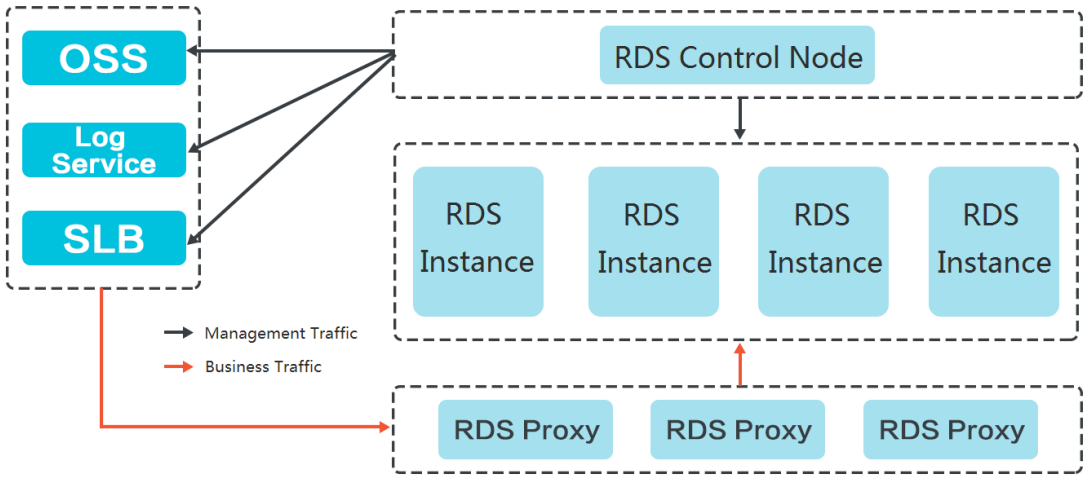
Originally based on a branch of MySQL, ApsaraDB RDS for MySQL provides excellent performance. It is a tried and tested solution that handled the high-volume concurrent traffic during Double 11. ApsaraDB RDS for MySQL provides basic features such as whitelist configuration, backup and restoration, Transparent Data Encryption (TDE), data migration, and management for instances, accounts, and databases. ApsaraDB RDS for MySQL also provides the following advanced features:

- **Read-only instance:** In scenarios where ApsaraDB RDS for MySQL handles a small number of write requests but a large number of read requests, you can create read-only instances to scale up the reading capability and increase the application throughput.
- **Read/write splitting:** The read/write splitting feature provides a read/write splitting endpoint. This endpoint enables an automatic link for the primary instance and all of its read-only instances. An application can connect to the read/write splitting endpoint to read and write data. Write requests are distributed to the primary instance and read requests are distributed to read-only instances based on their weights. To scale up the reading capability of the system, you can add more read-only instances.

## 3.2. Architecture

The following figure shows the system architecture of ApsaraDB RDS.

ApsaraDB RDS system architecture

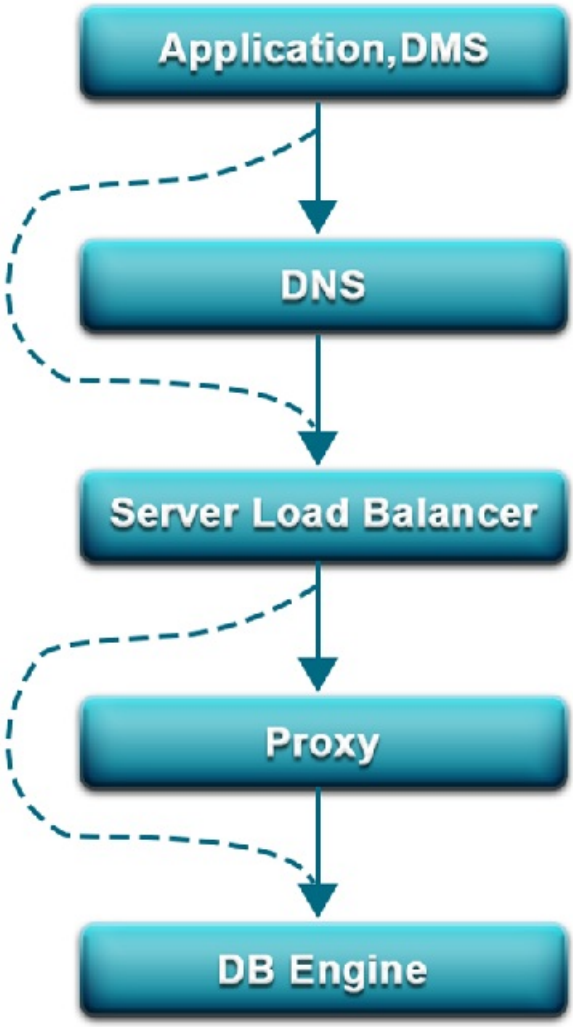



## 3.3. Features

### 3.3.1. Data link service

The data link service allows you to add, delete, modify, and query the table schema and data.

ApsaraDB RDS data link service




## DNS

The Domain Name System (DNS) module can dynamically resolve domain names to IP addresses. Therefore, IP address changes do not affect the performance of ApsaraDB RDS instances.

For example, the domain name of an ApsaraDB RDS instance is test.rds.aliyun.com, and its corresponding IP address is 10.1.1.1. The instance can be accessed when test.rds.aliyun.com or 10.1.1.1 is configured in the connection pool of a program.

After this ApsaraDB RDS instance is migrated or its version is upgraded, the IP address may change to 10.1.1.2. If the domain name test.rds.aliyun.com is configured in the connection pool, the instance can still be accessed. However, if the IP address 10.1.1.1 is configured in the connection pool, the instance is no longer accessible.

## SLB

The Server Load Balancer (SLB) module provides both the internal and public IP addresses of an ApsaraDB RDS instance. Therefore, server changes do not affect the performance of the instance.

For example, the internal IP address of an ApsaraDB RDS instance is 10.1.1.1, and the corresponding Proxy module or database engine runs on 192.168.0.1. The SLB module typically redirects all traffic destined for 10.1.1.1 to 192.168.0.1. If 192.168.0.1 fails, another server in the hot standby state with the IP address 192.168.0.2 takes over for the initial server. In this case, the SLB module redirects all traffic destined for 10.1.1.1 to 192.168.0.2, and the ApsaraDB RDS instance continues to provide services normally.

### Proxy

The Proxy module provides the following features:

- Data routing: aggregates the distributed complex queries in big data scenarios and provides the corresponding capacity management capabilities.
- Traffic detection: reduces SQL injection risks and supports SQL log backtracking when necessary.
- Session persistence: prevents database connection interruptions when faults occur.

### Database engines

The following table describes the major relational database management systems (RDBMSs) supported by ApsaraRD RDS.

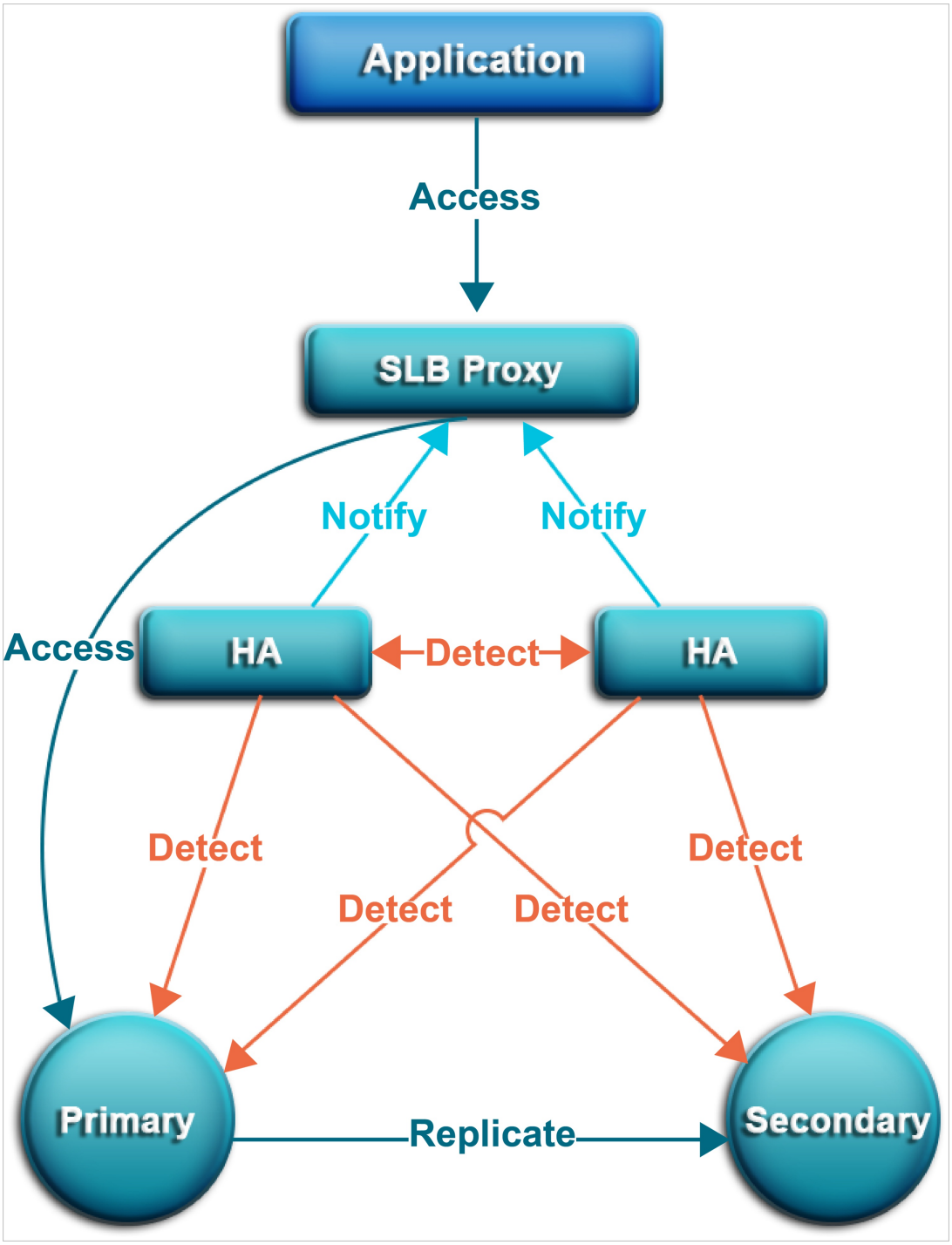
RDBMSs supported by ApsaraRD RDS

| RDBMS | Version |
| --- | --- |
| MySQL | 5.6 and 5.7 |
| PolarDB | 11 |

## 3.3.2. High-availability service

The high-availability (HA) service ensures the availability of data link services and processes internal database exceptions. The HA service is implemented by multiple HA nodes.

ApsaraDB RDS HA service




## Detection

The Detection module checks whether the primary and secondary nodes of the DB Engine are providing services normally.

The HA node uses heartbeat information taken at 8 to 10 second intervals to determine the health status of the primary node. This information, along with the health status of the secondary node and heartbeat information from other HA nodes, provides a reference for the Detection module. All this information helps the module avoid misjudgment caused by exceptions such as network jitter. Failover can be completed within a short time.

### Repair

The Repair module maintains the replication relationship between the primary and secondary nodes of the DB Engine. It can also correct errors that occur on the nodes during normal operations. For example:

- It can automatically restore primary/secondary replication after a disconnection.
- It can automatically repair table-level damage to the primary or secondary node.
- It can save and automatically repair the primary or secondary node when the node fails.

### Notice

The Notice module informs the Server Load Balancer (SLB) or Proxy module of status changes to the primary and secondary nodes to ensure that you always access the correct node.

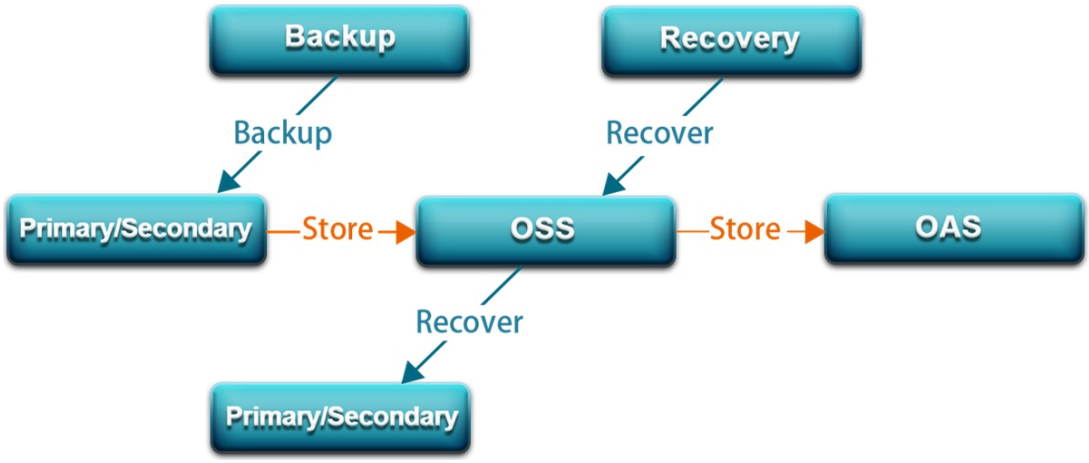
For example, the Detection module discovers problems with the primary node and instructs the Repair module to resolve these problems. If the Repair module fails to resolve a problem, it instructs the Notice module to perform traffic switchover. The Notice module forwards the switching request to the SLB or Proxy module. Then, all traffic is redirected to the secondary node.

Meanwhile, the Repair module creates a new secondary node on a different physical server and synchronizes this change back to the Detection module. The Detection module rechecks the health status of the instance.

## 3.3.3. Backup service

The backup service supports offline data backup, storage, and recovery.

ApsaraDB RDS backup service




### Backup

The Backup module compresses and uploads data and logs on both the primary and secondary nodes. ApsaraDB RDS uploads backup files to Object Storage Service (OSS) and dumps the backup files to a more cost-effective and persistent Archive Storage system. When the secondary node operates normally, backups are always created on the secondary node. This way, the services on the primary node are not affected. When the secondary node is unavailable or damaged, the Backup module creates backups on the primary node.

### Recovery

The Recovery module restores backup files from OSS to a destination node. The Recovery module provides the following features:

- Primary node rollback: rolls back the primary node to a specific point in time when an operation error occurs.
- Secondary node repair: creates a new secondary node to reduce risks when an irreparable fault occurs on the secondary node.
- Read-only instance creation: creates a read-only instance from backup files.

### Storage

The Storage module uploads, dumps, and downloads backup files.

All backup data is uploaded to OSS for storage. You can obtain temporary links to download the data.

In specific scenarios, the Storage module allows you to dump backup files from OSS to Archive Storage for more cost-effective and longer-term offline storage.

## 3.3.4. Monitoring service

ApsaraDB RDS provides multilevel monitoring services across the physical, network, and application layers to ensure service availability.

### Service

The Service module tracks the status of services. For example, the Service module monitors whether Server Load Balancer (SLB), Object Storage Service (OSS), and other cloud services on which ApsaraDB RDS depends are operating normally. The monitored metrics include functionality and response time. The Service module also uses logs to determine whether the internal services of ApsaraDB RDS are operating properly.

### Network

The Network module tracks statuses at the network layer. The following metrics are monitored:

- Connectivity between Elastic Compute Service (ECS) and ApsaraDB RDS
- Connectivity between physical servers of ApsaraDB RDS
- Rates of packet loss on vRouters and vSwitches

### OS

The OS module tracks the statuses of hardware and OS kernel. The following metrics are monitored:

- Hardware maintenance: The OS module constantly checks the operating status of the CPU, memory, motherboard, and storage device. It can predict faults in advance and automatically submit repair reports when it determines a fault is likely to occur.
- OS kernel monitoring: The OS module tracks all database calls and analyzes the causes of slow calls or call errors based on the kernel status.

### Instance

The Instance module collects the following information about ApsaraDB RDS instances:

- Instance availability information
- Instance capacity and performance metrics

- Instance SQL execution records

# 3.3.5. Scheduling service

The scheduling service allocates resources and manages instance versions.

## Resource

The Resource module allocates and integrates underlying ApsaraDB RDS resources when you activate and migrate instances. When you create an instance by using the ApsaraDB RDS console or an API operation, the Resource module calculates the most suitable host to carry traffic to and from the instance. A similar process occurs when ApsaraDB RDS instances are migrated.

After instances are repeatedly created, deleted, or migrated, the Resource module calculates the degree of resource fragmentation. In addition, it integrates resources on a regular basis to improve the service carrying capacity.

# 3.3.6. Migration service

The migration service can migrate data from your self-managed databases to ApsaraDB RDS.

## DTS

Data Transmission Service (DTS) can migrate data from your self-managed databases to ApsaraDB RDS without the need to stop services.

DTS is a data exchange service that streamlines data migration, real-time synchronization, and subscription. DTS is dedicated to implementing remote and millisecond-speed asynchronous data transmission in various scenarios. Based on the active geo-redundancy architecture designed for Double 11, DTS can make the data architecture secure, scalable, and highly available by providing real-time data streams to up to thousands of downstream applications.

# 4.Data Transmission Service (DTS)

## 4.1. What is DTS?

Data Transmission Service (DTS) is a data service that is provided by Alibaba Cloud. DTS supports data transmission between various types of data sources, such as relational databases.

### Features

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

| Feature | Description |
| --- | --- |
| Data migration | You can use DTS to migrate data between homogeneous and heterogeneous data sources. This feature applies to the following scenarios: data migration to Alibaba Cloud, data migration between instances within Alibaba Cloud, and database splitting and scale-out. |
| Data synchronization | You can use DTS to synchronize data between data sources. This feature applies to the following scenarios: disaster recovery, data backup, load balancing, cloud BI systems, and real-time data warehousing. |
| Change tracking | You can use DTS to track data changes from databases in real time. This feature applies to the following scenarios: cache updates, business decoupling, asynchronous data processing, synchronization of heterogeneous data, and synchronization of extract, transform, and load (ETL) operations. |

## 4.2. Environment requirements

You must use DTS on hosts of the following models:

- PF51.*
- PV52P2M1.*
- DTS_E.*
- PF61.*
- PF61P1.*
- PV62P2M1.*
- PV52P1.*
- Q5F53M1.*
- PF52M2.*
- Q41.*
- Q5N1.22
- Q5N1.2B
- Q46.22
- Q46.2B

- W41.22
- W41.2B
- W1.22
- W1.2B
- W1.2C
- D13.12

You must use the following operating system:

AliOS7U2-x86-64

> **Notice**
>
> - Do not use DTS on hosts that are excluded from the preceding models.
> - The */apsara* directory used by DTS resides on only one hard disk. Make sure that the available space in the directory is larger than 2 TB.
>
>   If the available space in the */apsara* directory is less than 2 TB, tasks cannot run as expected and errors will occur. If a task fails, the task recovery and data pulling are affected.

# 4.3. Benefits

Data Transmission Service (DTS) allows you to transfer data between various data sources, such as relational databases and online analytical processing (OLAP) databases. DTS provides the following data transmission methods: data migration, data synchronization, and change tracking. Compared with other data migration and synchronization tools, DTS provides transmission channels with higher compatibility, performance, security, and reliability. DTS also provides a variety of features to help you create and manage transmission channels.

## High compatibility

DTS allows you to migrate or synchronize data between homogeneous and heterogeneous data sources. For migration between heterogeneous data sources, DTS supports schema conversion.

DTS provides the following data transmission methods: data migration, data synchronization, and change tracking. In change tracking and data synchronization, data is transferred in real time.

DTS minimizes the impact of data migration on applications to ensure service continuity. The application downtime during data migration is minimized to several seconds.

## High performance

DTS uses high-end servers to ensure the performance of each data synchronization or migration channel.

DTS uses a variety of optimization measures for data migration.

Compared with traditional data synchronization, the data synchronization feature of DTS refines the granularity of concurrency to the transaction level. The feature allows you to synchronize incremental data in one table by using multiple concurrent channels. This improves synchronization performance.

## Security and reliability

DTS is implemented based on clusters. If a node in a cluster is unavailable or faulty, the control center switches all tasks on this node to another node in the cluster.

Secure transmission protocols and tokens are used for authentication across DTS modules to ensure reliable data transmission.

### Ease of use

The DTS console provides a codeless wizard for you to create and manage channels.

To facilitate channel management, the DTS console shows information about transmission channels, such as transmission status, progress, and performance.

DTS supports resumable transmission, and monitors channel status on a regular basis. If DTS detects a network failure or system error, DTS automatically fixes the failure or error and restarts the channel. If the failure or error persists, you must manually repair and restart the channel in the DTS console.

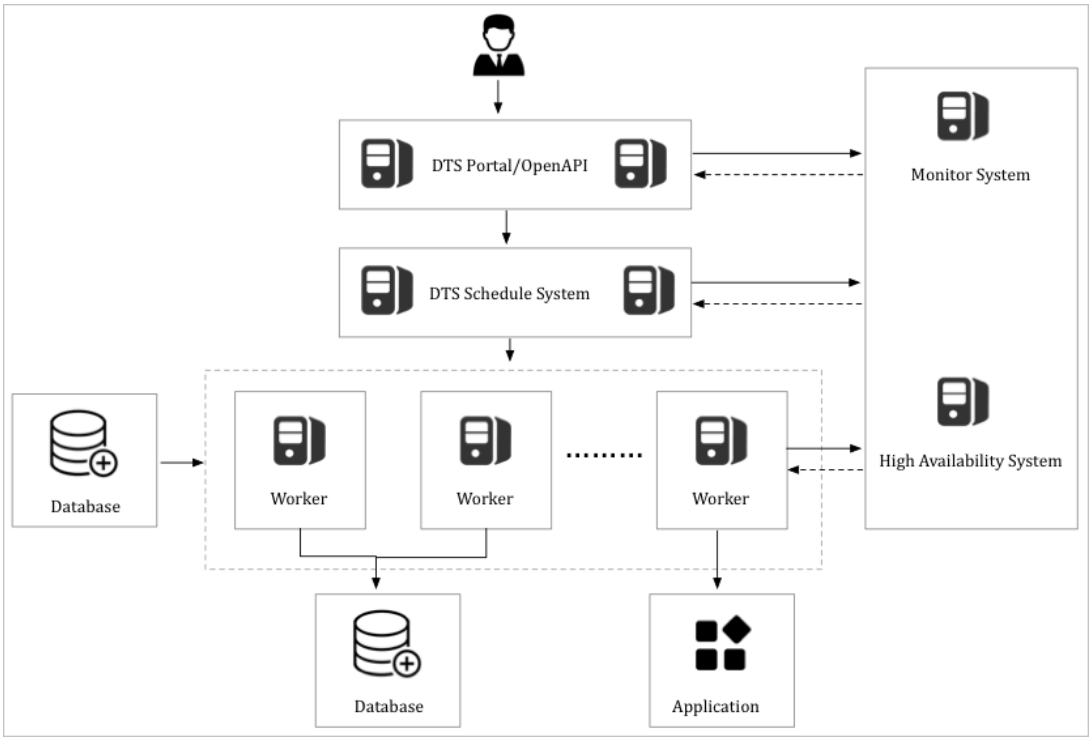
# 4.4. Architecture

This topic introduces the system architecture of Data Transmission Service (DTS) and the design concepts of DTS features.

### System architecture

The following figure shows the system architecture of Data Transmission Service (DTS).

System architecture




- High availability

  Each module in DTS has a primary node and a secondary node to ensure high availability. The disaster recovery module runs a health check on each node in real time. If a node failure is detected, the module switches the channel to a healthy node within only a few seconds.

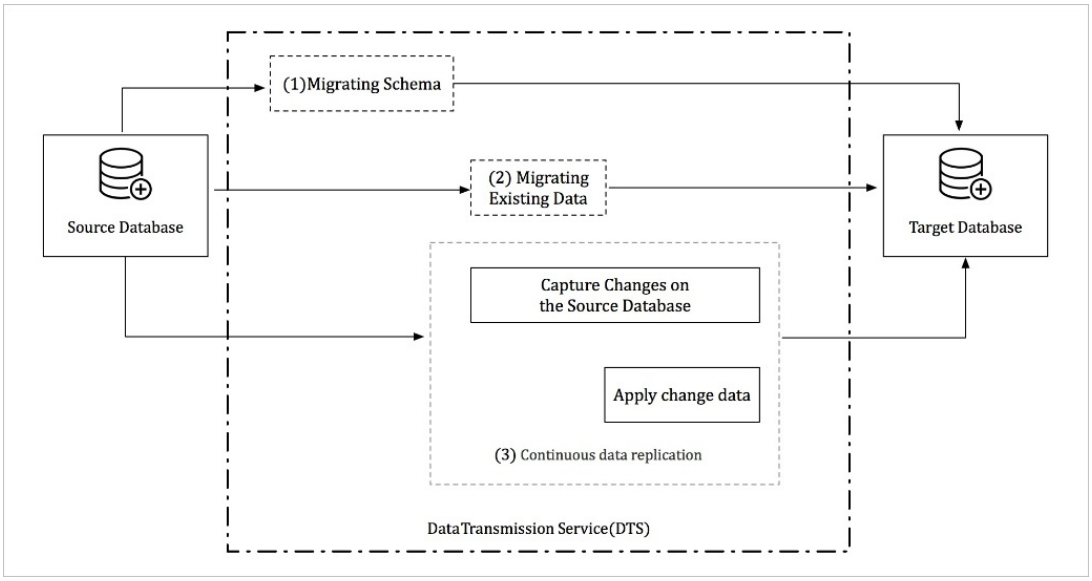
- Connection reliability

To ensure the connection reliability of change tracking and data synchronization channels, the disaster recovery module checks for configuration changes, such as changes of a data source address. If a data source address is changed, the module allocates a new connection method to ensure the stability of the channel.

## Design concept of data migration

The following figure shows the design concept of data migration.

Design concept of data migration




Data migration supports schema migration, full data migration, and incremental data migration. The following processes ensure service continuity during data migration:

1. Schema migration
2. Full data migration
3. Incremental data migration

To migrate data between heterogeneous databases, DTS reads the source database schema, converts the schema into the syntax of the destination database, and imports the schema to the destination database.

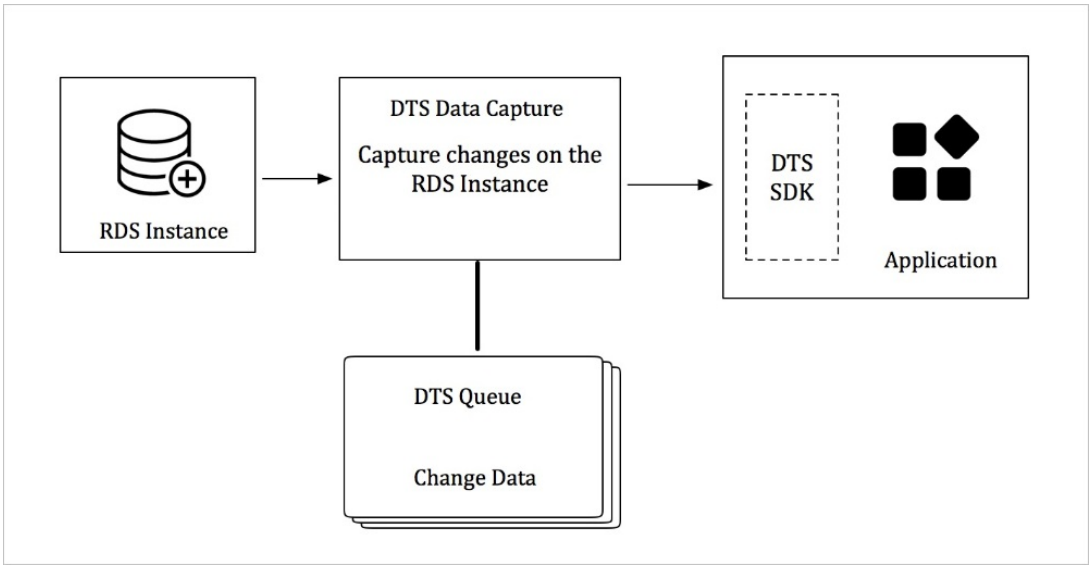
A full data migration requires a long period of time. During this process, incremental data is continuously written to the source database. To ensure data consistency, DTS starts the incremental data reading module before full data migration. This module retrieves incremental data from the source database, and parses, encapsulates, and locally stores the data.

After the full data migration is complete, DTS starts the incremental data loading module. This module retrieves incremental data from the incremental data reading module. After reverse parsing, filtering, and encapsulation, incremental data is migrated to the destination database in real time.

## Design concept of change tracking

The following figure shows the design concept of change tracking.

Design concept of change tracking




The change tracking feature allows you to obtain incremental data from an RDS instance in real time. You can subscribe to incremental data on the change tracking server by using DTS SDKs. You can also customize data consumption rules based on your business requirements.

The incremental data reading module on the server side of DTS retrieves raw data from the source instance. After parsing, filtering, and syntax conversion, incremental data is locally stored.

The incremental data reading module connects to the source instance by using a database protocol and retrieves incremental data from the source instance in real time. If the source instance is an ApsaraDB RDS for MySQL instance, the incremental data reading module connects to the source instance by using the binary log dump protocol.

DTS ensures high availability of the incremental data reading module and consumption SDK processes.

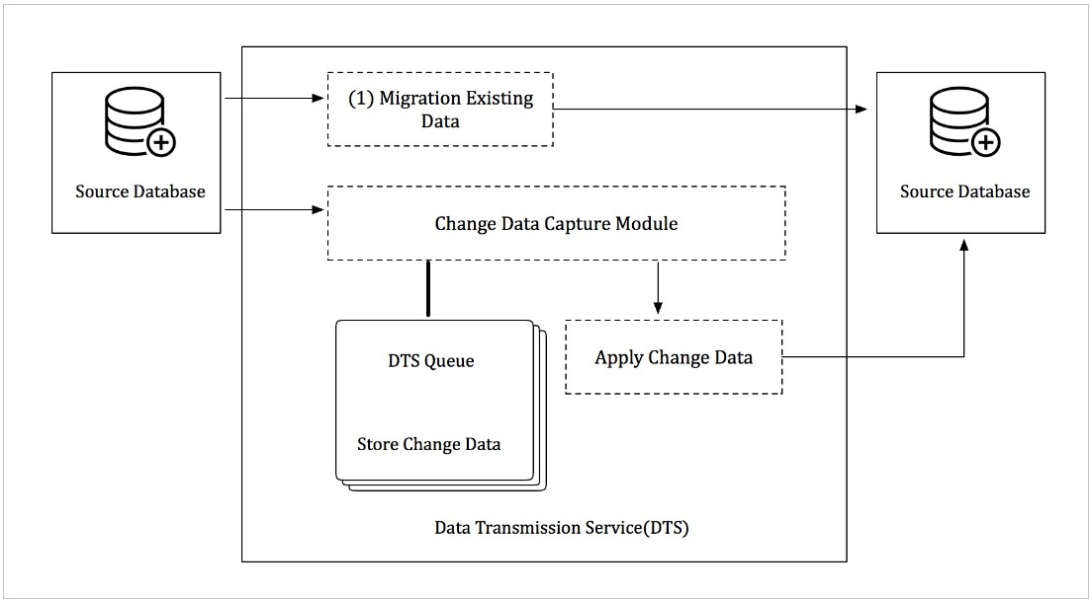
If an error is detected in the incremental data reading module, the disaster recovery module restarts the incremental data reading module on a healthy node. This ensures high availability of the incremental data reading module.

DTS ensures high availability of consumption SDK processes on the server. If you start multiple consumption SDK processes for the same change tracking channel, the server pushes incremental data to only one process at a time. If an error occurs on a process, the server pushes data to another healthy consumption process.

## Design concept of data synchronization

The following figure shows the design concept of data synchronization.

Design concept of data synchronization




The data synchronization feature can be used to synchronize incremental data between two RDS instances.

A data synchronization channel is established by using the following processes:

- Initial data synchronization: DTS synchronizes historical data from the source instance to the destination instance.
- Incremental data synchronization: After initial data synchronization, DTS synchronizes incremental data from the source instance to the destination instance.

DTS synchronizes incremental data by using the following modules:

- Incremental data reading module

  The incremental data reading module retrieves raw data from the source instance. After parsing, filtering, and syntax conversion, the data is locally stored. The incremental data reading module connects to the source instance by using a database protocol and obtains incremental data from the source instance. If the source instance is an ApsaraDB RDS for MySQL instance, the incremental data reading module connects to the source instance by using the binary log dump protocol.

- Incremental data loading module

  The incremental data loading module retrieves incremental data from the incremental data reading module and filters data based on the required objects. Then, the incremental data loading module synchronizes data to the destination instance without compromising transactional sequence and consistency.

DTS ensures high availability of the incremental data reading module and incremental data loading module. If a channel failure is detected, the disaster recovery module switches the channel to a healthy node. This ensures high availability of the synchronization channel.

# 4.5. Features

## 4.5.1. Data migration

You can use Data Transmission Service (DTS) to migrate data between various types of data sources. Typical scenarios include data migration to the cloud, data migration between instances within Apsara Stack, and database splitting and scale-out. Data migration supports the following extract, transform, and load (ETL) features: object name mapping and data filtering.

## Database and migration types

| Source database | Destination database | Migration type |
| --- | --- | --- |
| User-created MySQL database<br>Version 5.5, 5.6, 5.7, or 8.0 | User-created MySQL database<br>Version 5.5, 5.6, 5.7, or 8.0 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | User-created Kafka database<br>Version 0.10.1.0 to 1.0.2 | |
| User-created PostgreSQL database<br>Version 9.4, 9.5, 9.6, or 10.x | User-created PostgreSQL database<br>Version 9.4, 9.5, 9.6, or 10.x | |
| User-created Oracle database<br>Version 9i, 10g, 11g, 12c, 18c, or 19c | AnalyticDB for PostgreSQL<br>Version 4.3 or 6.0 | |
| User-created MongoDB database<br>Version 3.0, 3.2, 3.4, 3.6, or 4.0 | User-created MongoDB database<br>Version 3.0, 3.2, 3.4, 3.6, or 4.0 | • Full data migration<br>• Incremental data migration |
| User-created Redis database<br>Version 2.8, 3.0, 3.2, or 4.0 | User-created Redis database<br>Version 2.8, 3.0, 3.2, or 4.0 | Note<br>MongoDB and Redis are NoSQL databases that do not require schema migration. |

## Online migration

DTS uses online migration. You must configure the source instance, destination instance, and objects to be migrated. DTS automatically completes the entire data migration process. You can select all of the supported migration types to minimize the impact of online data migration on your services. However, you must ensure that DTS servers can connect to both the source and destination instances.

## Data migration types

DTS supports schema migration, full data migration, and incremental data migration.

- Schema migration: DTS migrates schemas from the source instance to the destination instance.
- Full data migration: migrates historical data from the source instance to the destination instance.
- Incremental data migration: DTS synchronizes incremental data that is generated during data migration from the source instance to the destination instance. You can select schema migration, full data migration, and incremental migration to migrate data with minimal downtime.

## ETL features

Data migration supports the following ETL features:

- Object name mapping: You can change the names of the columns, tables, and databases that are migrated to the destination database.
- Data filtering: You can use SQL conditions to filter the required data in a specific table. For example, you can specify a time range to migrate only the latest data.

## Alerts

If an error occurs during data migration, DTS immediately sends an SMS alert to the task owner. This allows the owner to handle the error at the earliest opportunity.

## Migration task

A migration task is a basic unit of data migration. To migrate data, you must create a migration task in the DTS console. To create a migration task, you must configure the required information such as the source and destination instances, migration types, and objects to be migrated. You can create, manage, stop, and delete migration tasks in the DTS console.

The following table describes the statuses of a migration task.

Task statuses

| Status | Description | Available operation |
| --- | --- | --- |
| Not Started | The migration task has been configured but no precheck is performed. | • Run a precheck<br>• Delete the migration task |
| Prechecking | A precheck is being performed but the migration task is not started. | Delete the migration task |
| Passed | The migration task has passed the precheck but has not been started. | • Start the migration task<br>• Delete the migration task |

| Status | Description | Available operation |
| --- | --- | --- |
| Migrating | The task is migrating data. | • Pause the migration task<br>• Stop the migration task<br>• Delete the migration task |
| Migration Failed | An error occurred during data migration. You can identify the point of failure based on the progress of the migration task. | Delete the migration task |
| Paused | The migration task is paused. | • Start the migration task<br>• Delete the migration task |
| Completed | The migration task is completed, or you have stopped data migration by clicking **End**. | Delete the migration task |

# 4.5.2. Data synchronization

You can use Data Transmission Service (DTS) to synchronize data between two data sources. This feature applies to various scenarios, such as data backup, disaster recovery, active geo-redundancy, cross-border data synchronization, load balancing, cloud BI systems, and real-time data warehousing.

## Supported databases

| Source database | Destination database | Initial synchronization type | Synchronization topology |
| --- | --- | --- | --- |
| • User-created MySQL database | User-created MySQL database<br>Version 5.5, 5.6, 5.7, or 8.0 | Initial schema synchronization<br>Initial full data synchronization | One-way synchronization<br>Two-way synchronization |

| Source database | Destination database | Initial synchronization type | Synchronization topology |
| --- | --- | --- | --- |
| • RDS MySQL<br>Version 5.5, 5.6, 5.7, or 8.0<br>• RDS MySQL<br>Version 5.6 or 5.7 | RDS MySQL<br>Version 5.6 or 5.7 | Initial schema synchronization<br>Initial full data synchronization | One-way synchronization<br>Two-way synchronization |
| PolarDB-X (formerly known as DRDS) | PolarDB-X | Initial full data synchronization | One-way synchronization |

## Objects to be synchronized

- You can select columns, tables, or databases as the objects to be synchronized. You can specify one or more tables that you want to synchronize.
- DTS allows you to synchronize data between tables that have different names, or between databases that have different names. You can use the object name mapping feature to specify the names of destination columns, tables, and databases.
- You can specify one or more columns that you want to synchronize.

## Synchronization tasks

A synchronization task is a basic unit of data synchronization. To synchronize data between two instances, you must create a synchronization task in the DTS console.

The following table describes the statuses of a synchronization task.

Task statuses

| Task status | Description | Available operation |
| --- | --- | --- |
| Prechecking | A precheck is being performed before the synchronization task is started. | • View the configurations of the synchronization task<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure monitoring and alerts |

| Task status | Description | Available operation |
| --- | --- | --- |
| Precheck Failed | The synchronization task has failed to pass the precheck. | • Run a precheck<br>• View the configurations of the synchronization task<br>• Reselect the objects to be synchronized<br>• Modify the synchronization speed<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure monitoring and alerts |
| Not Started | The synchronization task has passed the precheck but has not been started. | • Run a precheck<br>• Start the synchronization task<br>• Reselect the objects to be synchronized<br>• Modify the synchronization speed<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure monitoring and alerts |
| Performing Initial Synchronization | Initial synchronization is being performed. | • View the configurations of the synchronization task<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure monitoring and alerts |

| Task status | Description | Available operation |
| --- | --- | --- |
| Initial Synchronization Failed | The task has failed during initial synchronization. | • View the configurations of the synchronization task<br>• Reselect the objects to be synchronized<br>• Modify the synchronization speed<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure monitoring and alerts |
| Synchronizing | The task is synchronizing data. | • View the configurations of the synchronization task<br>• Reselect the objects to be synchronized<br>• Modify the synchronization speed<br>• Pause the synchronization task<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure monitoring and alerts |
| Synchronization Failed | An error occurred during synchronization. | • View the configurations of the synchronization task<br>• Reselect the objects to be synchronized<br>• Modify the synchronization speed<br>• Start the synchronization task<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure monitoring and alerts |

| Task status | Description | Available operation |
| --- | --- | --- |
| Paused | The synchronization task is paused. | • View the configurations of the synchronization task<br>• Reselect the objects to be synchronized<br>• Modify the synchronization speed<br>• Start the synchronization task<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure monitoring and alerts |

## Advanced features

You can use the following advanced features to facilitate data synchronization:

- Add or remove the objects to be synchronized

  You can add or remove the required objects when a task is synchronizing data.

- View and analyze the synchronization performance

  DTS provides trend charts that allow you to view and analyze the performance of your synchronization tasks. The synchronization performance is measured based on bandwidth, synchronization speed (RPS), and synchronization delay.

- Monitor synchronization tasks

  DTS allows you to monitor the status of synchronization tasks. If the threshold for synchronization delay is reached, you will receive an alert. You can set the alert threshold based on the sensitivity of your businesses to synchronization delays.

# 4.5.3. Change tracking

You can use Data Transmission Service (DTS) to track data changes from user-created MySQL databases in real time. This feature applies to the following scenarios: cache updates, business decoupling, synchronization of heterogeneous data, and synchronization of extract, transform, and load (ETL) operations.

## Supported databases

- User-created MySQL database
- User-created Oracle database

## Objects for change tracking

The objects for change tracking include tables and databases. You can specify one or more tables from which you want to track data changes.

In change tracking, data changes include data manipulation language (DML) operations and data definition language (DDL) operations. When you configure a change tracking task, you can select the operation type.

## Change tracking tasks

A change tracking task is the basic unit of change tracking and data consumption. To track data changes from a MySQL database, you must create a change tracking task for the MySQL database in the DTS console. The change tracking task pulls incremental data from the MySQL database in real time and locally stores the incremental data. You can use the DTS SDK to consume the incremental data from the change tracking task. You can also create, manage, or delete change tracking tasks in the DTS console.

A change tracking task can be consumed by only one downstream SDK client. To track data changes from a MySQL database by using multiple downstream SDK clients, you must create an equivalent number of change tracking tasks.

The Task statuses table describes the statuses of a change tracking task.

Task statuses

| Task status | Description | Available operation |
| --- | --- | --- |
| Prechecking | The change tracking task has been configured and a precheck is being performed. | Delete the change tracking task |
| Not Started | The change tracking task has passed the precheck but has not been started. | • Start the change tracking task<br>• Delete the change tracking task |
| Performing Initial Change Tracking | The initial change tracking is in progress. This process takes about 1 minute. | Delete the change tracking task |
| Normal | Incremental data is being pulled from the MySQL database. | • View the demo code<br>• View the tracked data<br>• Delete the change tracking task |
| Error | An error occurs when the change tracking task is pulling incremental data from the MySQL database. | • View the demo code<br>• Delete the change tracking task |

## Advanced features

You can use the following advanced features that are provided for change tracking:

- Add or remove the objects for change tracking

  You can add or remove the required objects when a change tracking task is running.

- View the tracked data

  You can view the data that is tracked by the change tracking task in the DTS console.

- Modify consumption checkpoints

  You can modify consumption checkpoints.

- Monitor change tracking tasks

  DTS allows you to monitor the status of change tracking tasks. If the threshold for consumption delay is reached, you will receive an alert. You can set the alert threshold based on the sensitivity of your businesses to consumption delays.

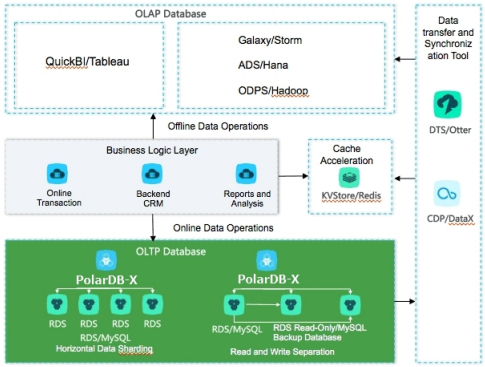
# 5.Cloud Native Distributed Database PolarDB-X

## 5.1. What is PolarDB-X?

Cloud Native Distributed Database PolarDB-X is a middleware service independently developed by Alibaba Group for scale-out of single-instance relational databases. It is compatible with Distributed Relational Database Service (DRDS).

PolarDB-X is the standard of relational database access for Alibaba Group. It shares the database sharding logic with Taobao Distributed Data Layer (TDDL). Compatible with the MySQL protocol, PolarDB-X supports most MySQL data manipulation language (DML) and data definition language (DDL) syntax. It provides the core capabilities of distributed databases, such as database sharding, table sharding, smooth scale-out, configuration changing, and transparent read/write splitting. It is lightweight (stateless), flexible, stable, and efficient, and provides you with O&M capabilities throughout the lifecycle of distributed databases.

PolarDB-X is mainly used for operations on large-scale online data, which focuses on frontend businesses for writing data to databases. By splitting data in specific business scenarios, it maximizes operation efficiency to meet the requirements of high-concurrency and low-latency database operations.




PolarDB-X mainly solves the following problems:

- Capacity bottleneck of single-instance databases: As the data volume and access volume increase, traditional single-instance databases encounter great challenges that cannot be completely solved by hardware upgrades. In distributed database solutions of PolarDB-X, multiple instances work jointly, which effectively resolves the bottlenecks of data storage capacity and access volumes.
- Difficult scale-out of relational databases: Due to the inherent attributes of distributed databases, data can be stored to different shards in PolarDB-X through smooth data migration, supporting the dynamic scale-out of relational databases.

## 5.2. Benefits

### Distributed architecture

The distributed architecture of Cloud Native Distributed Database PolarDB-X allows horizontal partitioning of data and the cluster deployment of a single service. In this way, single-instance bottlenecks of Server Load Balancer (SLB), PolarDB-X, and ApsaraDB RDS for MySQL are resolved and service scalability is achieved.

### High performance

PolarDB-X for RDS (MySQL) partitions data in specific business scenarios and clusters data based on major business operations, speeding up the response to online transactional operations. PolarDB-X for HiStore uses the columnar storage and knowledge grid to significantly speed up the response to common analytic operations such as large-scale data aggregation and ad hoc queries. It also helps reduce costs by achieving high compression ratio.

### Security

PolarDB-X supports an account and permission system similar to that of single-instance databases, and provides useful functions, such as the IP address whitelist and default disabling of high-risk SQL statements. It offers comprehensive API operations for support even if they need to be integrated into the local management system. We also provide complete product support and architecture services.

## 5.3. Architecture

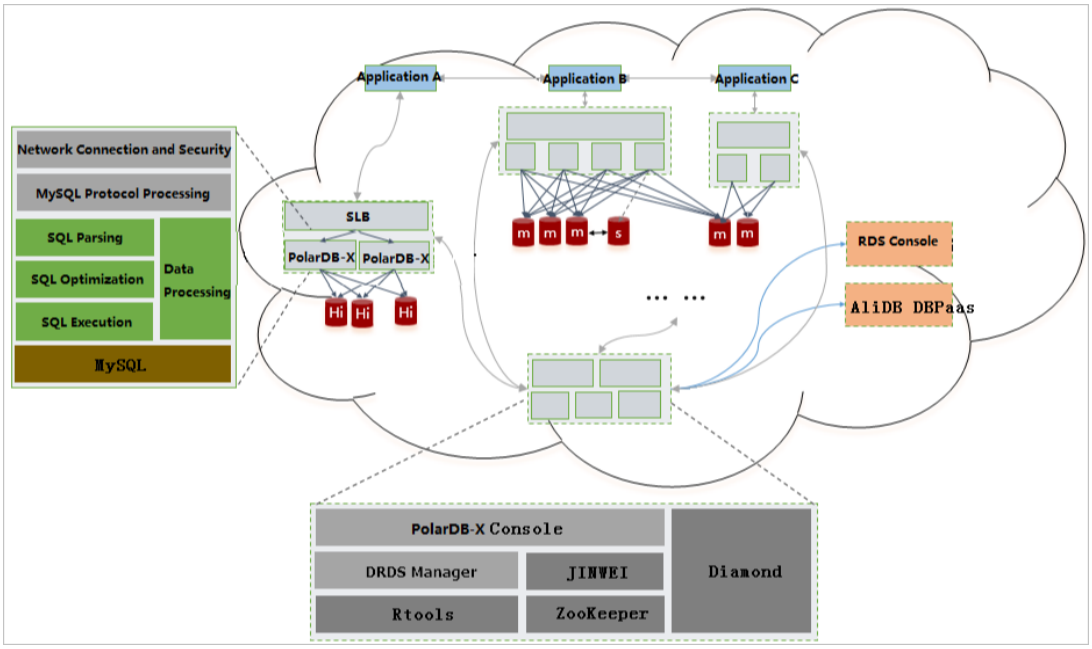
Cloud Native Distributed Database PolarDB-X (PolarDB-X) supports two data output methods: overall output by Apsara Stack and separate output by Alibaba middleware. The two output methods differ in features and dependent components of PolarDB-X.

The following table describes the differences between these two methods.

| Item | Overall output by Apsara Stack | Separate output by Alibaba middleware |
| --- | --- | --- |
| MySQL | ApsaraDB RDS for MySQL | Alibaba Cloud Database Platform as a Service (DBPaaS) |
| Load balancing | Centralized Server Load Balancer (Centralized SLB) | Client load balancer (VIPServer) |
| Special storage support | None | High compression-ratio column store (HiStore) |

The following figure shows the system architecture of PolarDB-X.

PolarDB-X system architecture




## PolarDB-X Server

PolarDB-X Server is the service layer of PolarDB-X. Multiple PolarDB-X Server nodes form a cluster to provide distributed database services, including read/write splitting, routed SQL execution, result merging, dynamic database configuration, and globally unique ID (GUID).

> **Note** PolarDB-X instances are stateless nodes. Therefore, ApsaraDB RDS for MySQL instances are used for storage. PolarDB-X implements data encryption by using encryption algorithms such as transparent data encryption (TDE) supported by ApsaraDB RDS for MySQL.

## High Availability Cluster

A redundancy design is adopted for each system component to prevent single point of failure (SPOFs).

## ApsaraDB RDS for MySQL (marked by "m" and "s" in the figure)

ApsaraDB RDS for MySQL stores data and performs data operations online. It implements high availability by using primary/secondary replication. It also implements dynamic database failover with the primary/secondary switchover mechanism.

You can implement management, monitoring, and alerting in the instance lifecycle in the ApsaraDB RDS for MySQL console.

## HiStore

When PolarDB-X outputs data separately (not overall output by Apsara Stack), it uses HiStore as the physical storage. HiStore is a low-cost, high-performance database developed by Alibaba to support column store. By using the column store, knowledge grid, and multiple cores, HiStore provides higher data aggregation and ad hoc query capabilities, with lower costs than row store (such as MySQL).

You can implement management, monitoring, and alerting within the instance lifecycle in the HiStore console.

## DBPaaS

When PolarDB-X outputs data separately (not overall output by Apsara Stack), the MySQL O&M platform DBPaaS implements management, monitoring, alerting, and resource management in the MySQL lifecycle.

### SLB

You do not need to install a client on user instances. SLB is used to distribute your requests. When an instance fails or a new instance is added, SLB ensures that traffic on the bound instances is distributed evenly.

### VIPServer

You must install a client on user instances, with a weak dependency on the central controller (interaction is performed only when the load configuration changes). VIPServer is used to distribute your requests. When an instance fails or a new instance is added, VIPServer ensures that traffic on the bound instances is distributed evenly.

### Diamond

Diamond is a system responsible for PolarDB-X configuration storage and management. It provides the configuration storage, query, and notification functions. Diamond stores the database source data, sharding rules, and PolarDB-X switch configuration.

### Data Replication System

Data Replication System is responsible for data migration and synchronization of PolarDB-X. The core capabilities of this system include full data migration and incremental data synchronization. Its derived features include smooth data import, smooth scale-out, and global secondary index (GSI). Data Replication System requires the support of ZooKeeper and PolarDB-X Rtools.

### PolarDB-X Console

PolarDB-X Console is designed for business database administrators (DBAs) to isolate resources as required and perform operations, such as instance management, database and table management, read/write splitting configuration, smooth scale-out, monitoring data display, and IP address whitelisting.

### PolarDB-X Manager

PolarDB-X Manager is designed for global O&M personnel and DBAs. It provides PolarDB-X resource management and system monitoring functions:

- Manages all resources on which ApsaraDB RDS for MySQL instances depend, including virtual machines, SLB instances, and domain names.
- Monitors the status of PolarDB-X instances, including queries per second (QPS), active threads, connections, node network I/O, and node CPU utilization.

### Rtools

Rtools is the O&M support system of PolarDB-X. It allows you to manage database configuration, read/write weight, connection parameters, database and table topologies, and sharding rules.

# 5.4. Features

# 5.4.1. Horizontal partitioning (sharding)

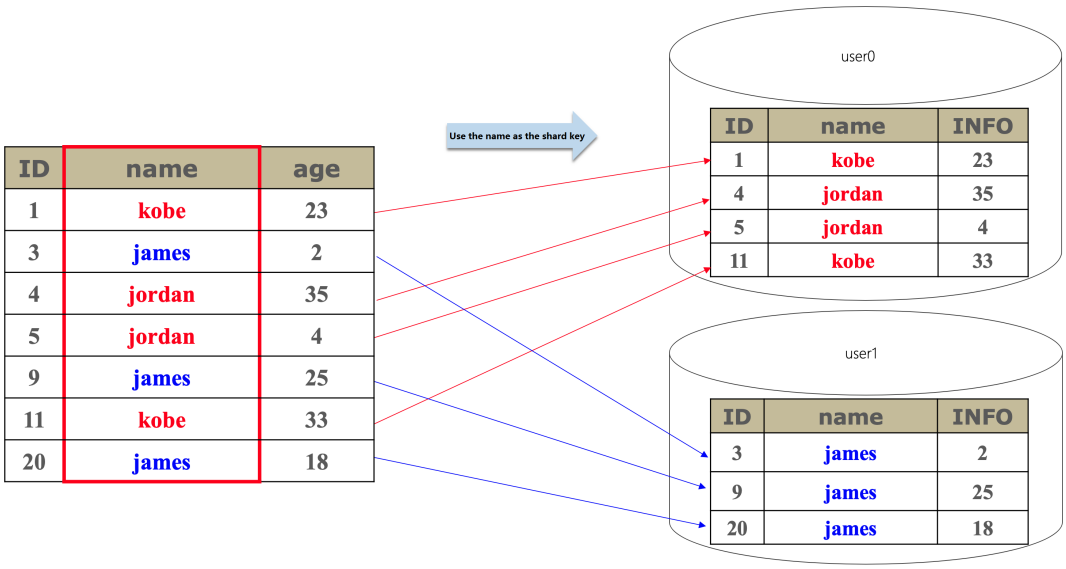
The core principle of PolarDB-X is horizontal partitioning of data, where data in a logical database is distributed and stored to multiple stable MySQL databases according to certain rules. These MySQL databases can be distributed across multiple instances or even across data centers, but provide external services (add, delete, modify, and query operations) as a single MySQL database. After partitioning, a physical database on an MySQL instance is called a database shard and a physical table is called a table shard (each table shard is a part of the complete data). By moving database shards on different MySQL instances, PolarDB-X implements database scale-out and improves the overall access to and the storage capacity of PolarDB-X databases.

PolarDB-X provides sharding rules, allowing you to select a partitioning policy that fits your business data characteristics. This ensures low latency for online database operations for transactions in high-concurrency scenarios. Therefore, when you use PolarDB-X, choosing the shard key is one of the important steps in database table structure design. The general principles are as follows:

- PolarDB-X performs well when writing data at the frontend. Most operations of such businesses are performed based on a specific database entity. For example, the business operations of the Internet are performed for users, the business operations of Internet of Things (IoT) are performed for devices and vehicles, the business operations of banks and government agencies are performed for customers, and the business operations of e-commerce independent software vendors (ISVs) and catering ISVs are performed for merchants. The data of such businesses can be partitioned by database entity. This, combined with global secondary indexes and eventually consistent transactions, can address the requirements on databases for large data volume, high concurrency, and low latency.
- For backend businesses, a batch of data is filtered and displayed on pages by condition and then processed and written back to the database. This is a business scenario in which PolarDB-X can partially address the needs. In this case, a large number of single-table associations and multi-table associations may exist, multiple filtering conditions are combined for DELETE and SELECT operations, and a large number of multi-table transactions are processed. Data partitioning by entity is recommended for such scenarios. If database processing is tightly related to time, data can be partitioned by time.

The following figure shows how data partitioning works.

Figure of data partitioning



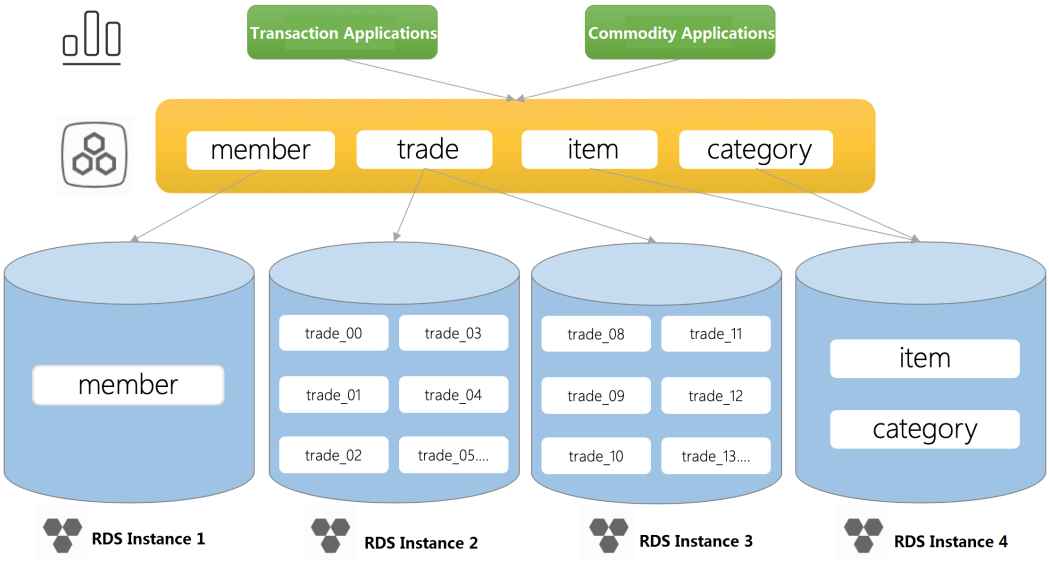
# 5.4.2. Vertical partitioning

The vertical partitioning capability of Distributed Relational Database Service (DRDS) aggregates business databases in different ApsaraDB RDS for MySQL instances to one connection. Through one connection, businesses can operate data in different databases, and perform joint queries and write transactions across databases in multiple ApsaraDB RDS for MySQL databases.

Vertical partitioning has business semantics. For example, tables related to transactions make up a transaction database, tables related to users make up a user database, and tables related to commodity make up a commodity database, which are placed in different ApsaraDB RDS for MySQL instances. Through vertical partitioning, databases on different instances can be aggregated to one connection and data in different databases can be operated through this connection.

The following figure shows how vertical partitioning works.

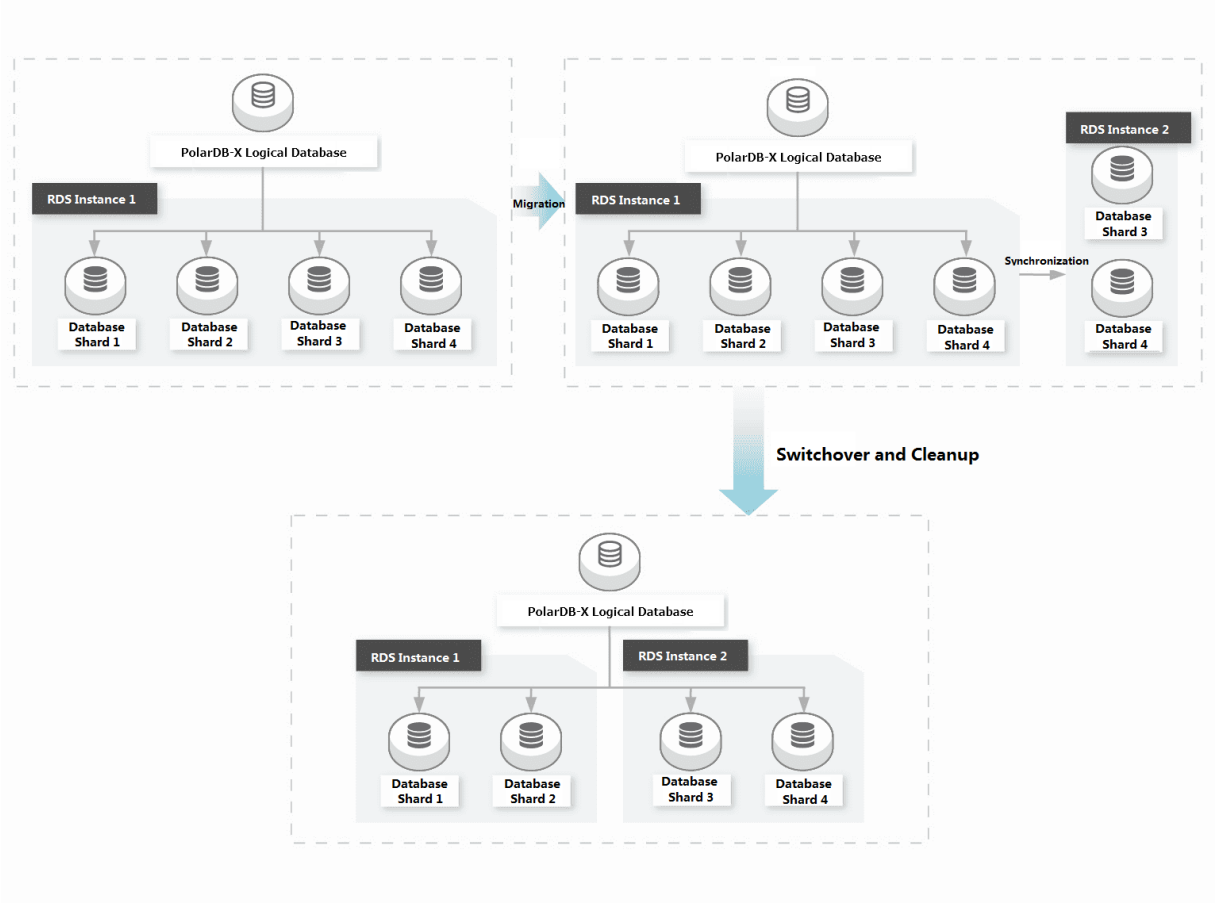
Vertical partitioning




# 5.4.3. Smooth scale-out

To scale out a PolarDB-X instance, you can add ApsaraDB RDS for MySQL instances and migrate the original database shards to the new ApsaraDB RDS for MySQL instances.

Smooth scale-out is an online horizontal expansion method. It smoothly migrates the original database shards to the new ApsaraDB RDS for MySQL instances and increases the overall data storage capacity by adding ApsaraDB RDS for MySQL instances, which reduces the pressure on each RDS instance to process data.




## How PolarDB-X scale-our works

Follow these steps:

1. Create a scale-out plan.

   Select a new ApsaraDB RDS for MySQL instance and database shards to be migrated. After the task is submitted, the system automatically creates a database and an account on the destination instance and submits a task for data migration and synchronization.

2. Perform full data migration.

   The system selects a time point before the current time and copies and migrates all data generated before this time point.

3. Perform incremental synchronization.

   After a full migration is completed, incremental data is synchronized according to the incremental change logs generated between a time point before the full migration and the current time, and eventually, the data is synchronized from the source database shard to the destination database shard in real time.

4. Verify data.

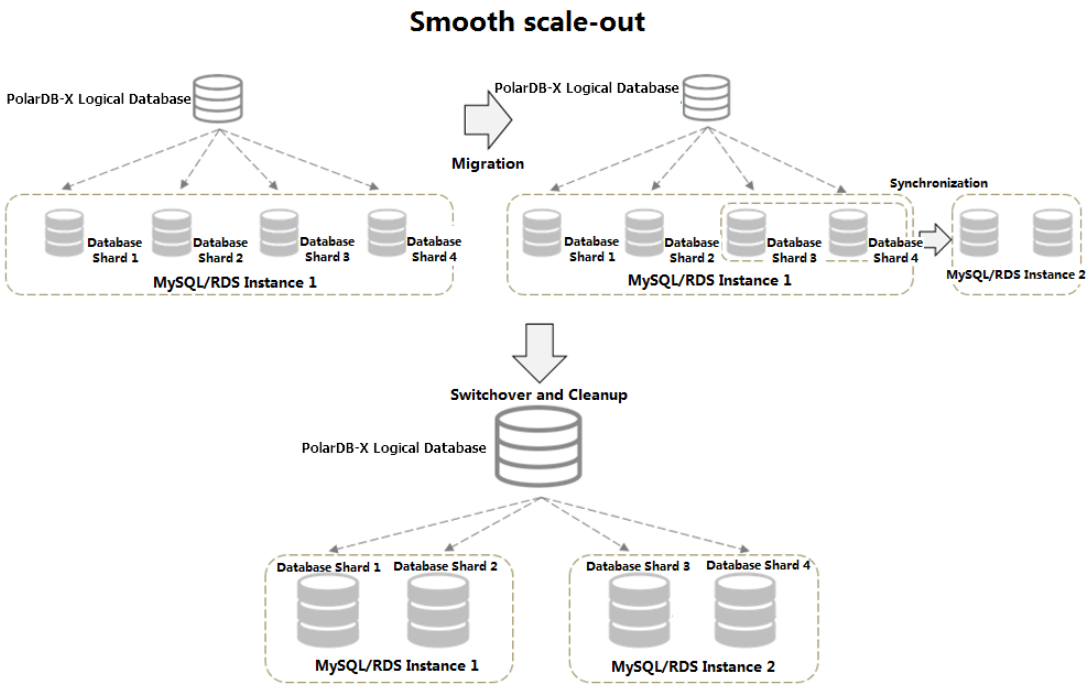
   When the incremental data is synchronized in quasi-real time, the system automatically performs full data verification and corrects inconsistent data caused by synchronization latency.

5. Disable the application service and switch routes.

After verification, the incremental data is still synchronized in quasi-real time, and a specified time is selected for the switch. To ensure strict data consistency, we recommend that you disable the service (you can also not disable the service but the same data may be overwritten at a high concurrency). The engine layer switches routes based on database sharding rules to switch subsequent traffic to the new database. The switching process can be completed within seconds.

The following figure shows data migration between database shards.

Scale-out




To ensure data security and facilitate rollback of a scale-out task, data synchronization continues after the routing rule is switched. After the data O&M personnel confirm that the service is normal, you can clean up data in the source database shard in the console.

The whole scale-out process has little impact on services of the upper layer (some services may be affected if the instance type of the ApsaraDB RDS for MySQL instance is not satisfactory or its traffic pressure is high). If the service is not disabled during the switch, we recommend that you perform this operation when the database access traffic is low to reduce the possibility of concurrently updating the same data.

# 5.4.4. Read/write splitting

The read/write splitting function of PolarDB-X is a relatively transparent policy to switch over the read traffic for ApsaraDB RDS for MySQL instances.

You can add read-only ApsaraDB RDS for MySQL instances and adjust their read weights in the PolarDB-X console without code modification if your business applications can tolerate the latency of data synchronization between read-only instances and the primary instance. The read traffic is proportionally adjusted between the primary ApsaraDB RDS for MySQL instance and multiple read-only ApsaraDB RDS for MySQL instances. Write operations and transaction operations are performed on the primary ApsaraDB RDS for MySQL instance.

Note that a latency exists for data synchronization between the primary instance and read-only instances. When a large data definition language (DDL) statement is executed or a large volume of data is being corrected, the latency may be over one minute. Therefore, consider whether your business can tolerate the impact before using this function.

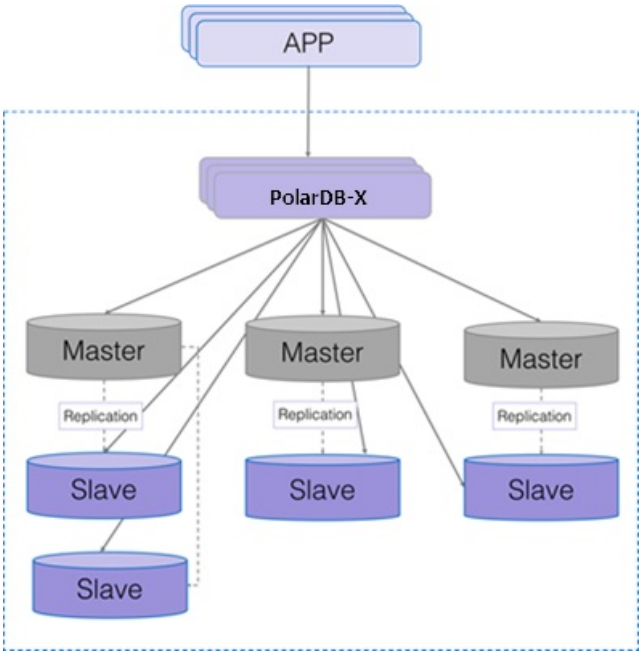
Adding read-only instances improves the read performance linearly. For example, if there is only one read-only instance, the read performance is doubled after one other read-only instance is added or tripled after two other read-only instances are added.

## Traffic distribution and instance addition for read/write splitting

The read/write splitting function of PolarDB-X requires no modification of application code. You only need to add read-only instances and adjust the weights of read operations in the PolarDB-X console, to proportionally adjust the read traffic between the primary instance and multiple read-only instances. The write operations are performed on the primary instance.

Adding read-only instances improves the read performance linearly. For example, if there is one read-only instance, the read performance is doubled after one other read-only instance is added or tripled after two other read-only instances are added, as shown in the following figure.

Traffic distribution and expansion for read/write splitting




All data in the read operations on a read-only instance is asynchronously synchronized from the primary instance with a millisecond-level latency. For SQL statements that require high real-time performance, you can specify the primary instance through PolarDB-X Hint to execute these SQL statements, as shown in the following code:

```
/*TDDL:MASTER/select * from tddl5_users;
```

PolarDB-X allows you to run SHOW NODE to view the actual distribution of read traffic, as shown in the following figure.

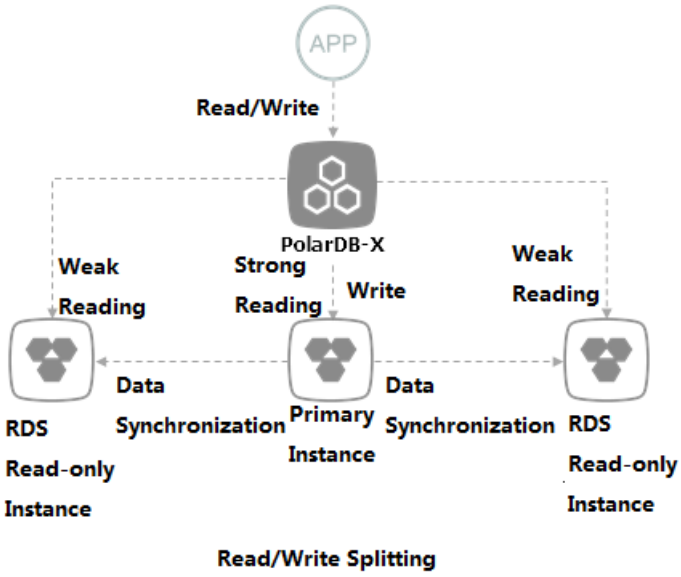
SHOW NODE to view the actual distribution of read traffic

```
mysql> show node;
+------+------------------+-------------------+------------------+---------------------+---------------------+
| ID   | NAME             | MASTER_READ_COUNT | SLAVE_READ_COUNT | MASTER_READ_PERCENT | SLAVE_READ_PERCENT |
+------+------------------+-------------------+------------------+---------------------+---------------------+
| 0    | USERDATABASE_RDS |                10 |                2 |                 83% |                 17% |
+------+------------------+-------------------+------------------+---------------------+---------------------+
1 row in set (0.00 sec)
```

## Support for transactions by read/write splitting

Read/write splitting is valid only for read requests (query requests) that are not in explicit transactions (transactions that need to be explicitly committed or rolled back). Write requests and read requests (including read-only transactions) in explicit transactions are executed in the primary instance and are not distributed to read-only instances.

- Common SQL statements for read requests include SELECT, SHOW, EXPLAIN, and DESCRIBE.
- Common SQL statements for write requests include INSERT, REPLACE, UPDATE, DELETE, and CALL.




## Read/write splitting in non-partition mode

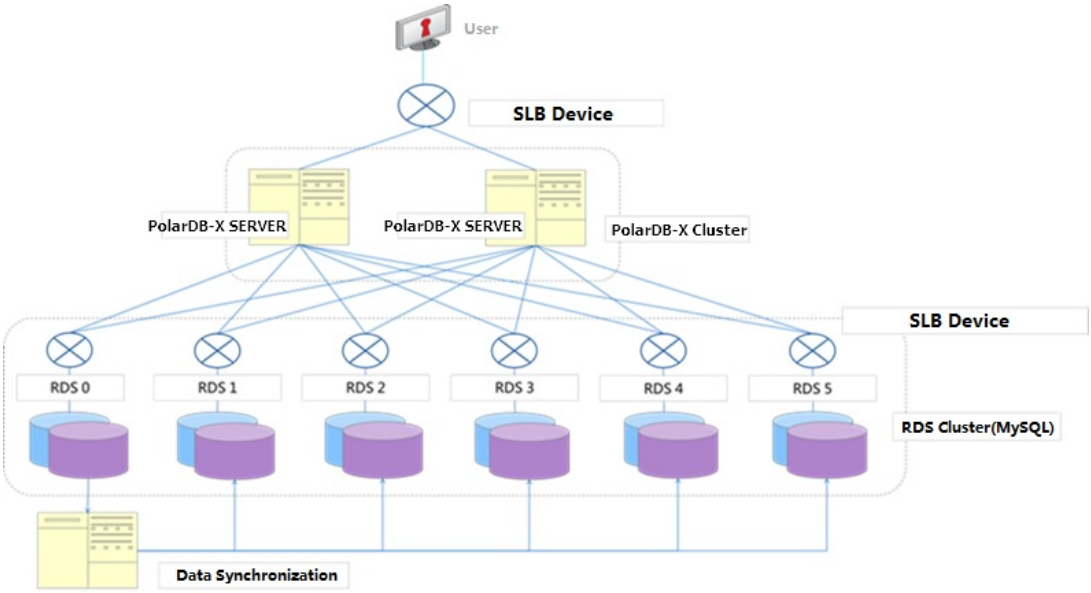
The read/write splitting function of PolarDB-X can be used independently in non-partition mode.

When you select an ApsaraDB RDS for MySQL instance for creating a PolarDB-X database in the PolarDB-X console, you can directly introduce a logical database on the ApsaraDB RDS for MySQL instance to the PolarDB-X database for read/write splitting without data migration.

# 5.4.5. Service upgrade and downgrade

A PolarDB-X instance consists of multiple server nodes that are deployed in a cluster and provides services externally through Server Load Balancer (SLB) and Domain Name System (DNS). The server nodes of PolarDB-X do not synchronize states, they process external requests in a balanced manner. When the processing capability of the server cluster is insufficient, server nodes can be added in real time to improve the service capability. If the resource utilization of all PolarDB-X server nodes in the cluster is low, you can remove some server nodes to downsize the cluster, and lower the capability of the service layer, for the elastic scaling of service capabilities. Figure of service upgrade and downgrade shows the details.

Figure of service upgrade and downgrade




# 5.4.6. Account and permission system

The account and permission system of PolarDB-X is used in the same way as MySQL, but does not support authorization across multiple databases, and has fewer permissions than MySQL. The system supports statements including GRANT, REVOKE, SHOW GRANTS, CREATE USER, DROP USER, and SET PASSWORD. It can be used to grant database-level and table-level permissions, but global and column-level permissions are not supported.

Account rules:

- The administrator account created in the console has all permissions.
- Only the administrator account can create and authorize accounts. Other accounts can only be created and authorized by the administrator account.
- The administrator account is bound to a database and does not have permissions on other databases. It can only access the bound database, and cannot grant permissions of other databases to an account. For example, the easydb administrator account can only connect to the easydb database, and can only grant permissions of the easydb database or tables in the easydb database to an account.

Currently, eight table-associated basic permissions are supported: CREATE, DROP, ALTER, INDEX, INSERT, DELETE, UPDATE, and SELECT. Among these operations:

- The TRUNCATE operation requires the table-level DROP permission.
- The REPLACE operation requires the table-level INSERT and DELETE permissions.

- The CREATE INDEX and DROP INDEX operations require the table-level INDEX permission.
- The CREATE SEQUENCE operation requires the database-level CREATE permission.
- The DROP SEQUENCE operation requires the database-level DROP permission.
- The ALTER SEQUENCE operation requires the database-level ALTER permission.
- The INSERT ON DUPLICATE UPDATE statement requires the table-level INSERT and UPDATE permissions.

# 5.4.7. PolarDB-X sequence

The PolarDB-X sequence of PolarDB-X (a 64-digit number of the signed BIGINT type in MySQL) aims to generate a globally unique number sequence (not necessarily in increments). This sequence is usually used to generate keys such as a primary key column and a unique key.

The PolarDB-X sequence of PolarDB-X can be implicitly used. When a table is partitioned to table shards, the primary key is auto_increment, and business data is inserted with no primary key specified, the globally unique primary key is automatically set, just like the single-instance MySQL database.

The PolarDB-X sequences can also be explicitly used. You can run `select xxx_seq.nextval from dual where count=?` to obtain one or more PolarDB-X sequences for other use in the application.

# 5.4.8. Second-level monitoring

PolarDB-X allows users to run **SHOW FULL STATS** to implement second-level monitoring. This command operates with the business monitoring system of PolarDB-X or third-party open-source monitoring software to provide better monitoring and alerting effect.

The following table describes the metrics supported by this command.

| Metric | Description |
| --- | --- |
| QPS | The logical queries per second (QPS) specifying queries from an application to a PolarDB-X instance. |
| RDS_QPS | The QPS from a PolarDB-X instance to an ApsaraDB RDS for MySQL instance is called physical QPS. |
| ERROR_PER_SECOND | The number of errors per second, which is the sum of SQL syntax errors, primary key conflicts, system errors, and connectivity errors. |
| VIOLATION_PER_SECOND | The number of primary key conflicts or unique key conflicts per second. |
| MERGE_QUERY_PER_SECOND | The number of queries merged from multiple table shards. |
| ACTIVE_CONNECTIONS | The number of connections in use. |
| CONNECTION_CREATE_PER_SECOND | The number of connections created per second. |
| RT(MS) | The logical response time (RT) for a response from an application to a PolarDB-X instance. |

| Metric | Description |
| --- | --- |
| RDS_RT(MS) | The physical RT for a response from a PolarDB-X instance to an ApsaraDB RDS for MySQL instance. |
| NET_IN(KB/S) | The network traffic received by PolarDB-X |
| NET_OUT(KB/S) | The network traffic sent by PolarDB-X |
| THREAD_RUNNING | The number of running threads. |
| HINT_USED_PER_SECOND | The number of queries with hints per second. |
| HINT_USED_COUNT | The total number of queries with hints since startup. |
| AGGREGATE_QUERY_PER_SECOND | The number of aggregate queries per second. |
| AGGREGATE_QUERY_COUNT | The total number of historical aggregate queries. |
| TEMP_TABLE_CREATE_PER_SECOND | The number of temporary tables created per second. |
| TEMP_TABLE_CREATE_COUNT | The total number of temporary tables created since startup. |
| MULTI_DB_JOIN_PER_SECOND | The number of cross-database JOIN queries per second. |
| MULTI_DB_JOIN_COUNT | The total number of cross-database JOIN queries since startup. |

# 5.4.9. Distributed SQL engine

The distributed SQL engine of PolarDB-X is designed to achieve high compatibility with a single-instance MySQL database, to implement SQL push-down. PolarDB-X allows you to perform SQL operations, such as analysis, optimization, routing, and data aggregation.

The core principles of SQL push-down are as follows:

- Process data as close to the data as possible.
- Reduce data transfers over the network.
- Reduce data processing on PolarDB-X and offload the processing work to the lower-level data nodes whenever possible.
- Make full use of the features and capabilities of database storage.

# 5.4.10. High-availability architecture

## Automatic traffic switchover of PolarDB-X Server

The PolarDB-X Server component of a PolarDB-X instance consists of multiple server nodes and provides services as a single connection through a load balancing service. When a PolarDB-X server fails, its traffic switches over to another PolarDB-X server in seconds. The entire failover process is transparent to users, with no need to change the application code or restart the application.

### Automatic traffic switchover

PolarDB-X supports the read/write splitting function. You can sign in to the console, choose **DRDS Database > Read/Write Splitting**, and configure the function. This function allocates some read traffic to the secondary instances. PolarDB-X identifies the read SQL requests and distributes them to the primary and secondary ApsaraDB RDS for MySQL instances based on the configured ratio, to implement read/write splitting. A secondary instance allocated with only read traffic is called a read-only instance.

If multiple read-only instances are configured but one of them fails (the connection fails), PolarDB-X automatically withdraws the read traffic from the failed instance, and then re-allocates the traffic based on the ratio of read traffic in the remaining normal read-only instances.

The automatic traffic switchover process of read-only instances is transparent to users. You do not need to restart applications. When no read-only instance is available, read requests are still allocated proportionally to both the read-only and the primary instances, to prevent the primary instance from being overloaded. Errors are reported for the read requests allocated to read-only instances .

> **Note** All write requests and transactions are automatically routed to the primary instance for execution, regardless of the availability of read-only instances.

## 5.4.11. Software upgrade

- PolarDB-X automatically provides new versions of installed database software.
- Software upgrade is optional. It is carried out only upon your request.
- If PolarDB-X determines that your version has major security risks, it will notify you to schedule the upgrade. The PolarDB-X team will provide support during the entire upgrade process.
- The PolarDB-X upgrade process is generally completed within 5 minutes. During the upgrade process, there may be several transient database disconnections. There is minimal interruption to applications if the database reconnection (or connection pool) is properly configured for applications.

## 5.4.12. SQL compatibility

PolarDB-X is compatible with the MySQL protocols and supports most MySQL query syntax, common data manipulation language (DML) syntax, and data definition language (DDL) syntax. However, the pronounced architectural differences between distributed databases and single-instance databases restrict the usage of SQL. The compatibility and SQL restrictions are described as follows.

> **Note** Since there are many MySQL versions and the MySQL syntax and PolarDB-X versions are constantly updating, the compatibility discussed in this document is for reference only. Determine whether the selected MySQL version matches your business according to the actual test results.

### PolarDB-X SQL restrictions

SQL restrictions are as follows:

- Custom data types and functions are not supported at present.
- Views, stored procedures, triggers, and cursors are not supported at present.
- Compound statements such as `BEGIN...END, LOOP...END LOOP, REPEAT...UNTIL...END REPEAT, and WHILE...DO...END WHILE` are not supported at present.

- Process control statements such as IF and WHILE are not supported at present.

## Small syntax restrictions

DDL:

- CREATE TABLE tbl_name LIKE old_tbl_name does not support table sharding.
- CREATE TABLE tbl_name SELECT statement does not support table sharding.

DML:

- SELECT INTO OUTFILE, SELECT INTO DUMPFILE, and SELECTvar_name are not supported at present.
- INSERT DELAYED is not supported at present.
- Subqueries irrelevant to the WHERE condition are not supported at present.
- SQL subqueries that contain aggregation conditions are not supported at present.
- Variable references and operations in SQL statements are not supported at present, for example, `SET @c=1, @d=@c+1; SELECT @c, @d`.

Database management:

- SHOW WARNINGS does not support the LIMIT/COUNT combination.
- SHOW ERRORS does not support the LIMIT/COUNT combination.

## Compatibility of PolarDB-X with SQL

Compatibility with MySQL protocols

PolarDB-X supports mainstream clients such as MySQL Workbench, Navicat For MySQL, and SQLyog.

> **Note** PolarDB-X supports the add, delete, modify, and query operations on databases. However, other special functions (such as import and diagnosis) have not been thoroughly tested.

PolarDB-X is compatible with the following DDL statements:

- `CREATE TABLE`
- `CREATE INDEX`
- `DROP TABLE`
- `DROP INDEX`
- `ALTER TABLE`
- `TRUNCATE TABLE`

PolarDB-X is compatible with the following DML statements:

- INSERT
- REPLACE
- UPDATE
- DELETE
- Subquery
- Scalar subquery
- Comparisons subquery
- Subquery with ANY, IN, or SOME
- Subquery with ALL

- Subquery by column
- Subquery with EXISTS or NOT EXISTS
- Subquery in the FROM clause
- SELECT

PolarDB-X is compatible with the following PREPARE statements:

- PREPARE
- EXECUTE
- DEALLOCATE PREPARE

PolarDB-X is compatible with the following database management statements

- SET
- SHOW
- KILL 'PROCESS_ID' (PolarDB-X only supports the KILL 'PROCESS_ID' command but does not support the KILL QUERY command.)
- SHOW COLUMNS
- SHOW CREATE TABLE
- SHOW INDEX
- SHOW TABLES
- SHOW TABLE STATUS
- SHOW TABLES
- SHOW VARIABLES
- SHOW WARNINGS
- SHOW ERRORS

> **Notice** Other SHOW commands are delivered to the database for processing by default, and the returned result data in different shards are not merged.

PolarDB-X is compatible with the following database tool statements:

- DESCRIBE
- EXPLAIN
- USE

Custom instructions of PolarDB-X are as follows:

- SHOW SEQUENCES, CREATE SEQUENCE, ALTER SEQUENCE, and DROP
- SEQUENCE. It manages PolarDB-X sequences.
- SHOW PARTITIONS FROM TABLE. It queries table shard keys.
- SHOW TOPOLOGY FROM TABLE. It queries the physical topology of a table.
- SHOW BROADCASTS. It queries all broadcast tables.
- SHOW RULE [FROM TABLE]. It queries the table sharding rule.
- SHOW DATASOURCES. It queries data sources of the backend database connection pool.
- SHOW DBLOCK/RELEASE DBLOCK. It defines the distributed LOCK.
- SHOW NODE. It queries the database read and write traffic.

- SHOW SLOW. It queries the slow SQL statements.
- SHOW PHYSICAL_SLOW. It queries slow SQL statements executed in the physical database.
- TRACE SQL_STATEMENT/SHOW TRACE. It traces the SQL statement execution process.
- EXPLAIN [DETAIL/EXECUTE] SQL_STATEMENT. It analyzes the SQL execution plans of PolarDB-X and physical databases.
- RELOAD USERS. It synchronizes the user information from PolarDB-X Console to the PolarDB-X Server.
- RELOAD SCHEMA. It clears data caches in the corresponding PolarDB-X database, such as cache of SQL parsing, syntax tree, and table structure.
- RELOAD DATASOURCES. It rebuilds a connection pool that connects the backend to all databases.

Database functions:

- SQL statements with shard keys are supported by all MySQL functions.
- SQL statements without a shard key are supported by only some functions.
- Operator functions

| Function | Description |
| --- | --- |
| AND, && | Logical AND |
| = | Assigns a value (a part of the SET statement or a part of the SET clause in the UPDATE statement). |
| BETWEEN... AND... | Determines a certain range of a value. |
| BINARY | Converts a string into a binary string. |
| & | Bitwise AND |
| ~ | Bitwise negation |
| ^ | Bitwise Exclusive OR (XOR) |
| DIV | Returns an integer obtained from integer division. |
| / | Division operator |
| <=> | NULL-safe equal operator |
| = | Equal operator, it compares the equality of two strings |
| >= | Operator, greater than or equal to |
| > | Greater than |
| IS NOTNULL | Tests for a non-NULL value. |
| ISNOT | Tests for a non-Boolean value. |
| ISNULL | It tests for a NULL value. |
| IS | Tests for a Boolean value. |

| Function | Description |
| --- | --- |
| << | Bitwise left shift operator |
| <= | Operator, less than or equal to |
| < | Operator, less than |
| LIKE | Compares a character string to a specified string pattern. |
| - | Minus operator |
| %, | Returns the remainder of a number divided by another number. |
| NOT BETWEEN... AND... | Determines a certain range that a value is not in. |
| ! =, <> | Operator, not equal to |
| NOT LIKE | Finds a specific character string that does not match a specified pattern. |
| NOT REGEXP | NOT operator in regular expressions |
| NOT, ! | NOT |
| OR | Logical OR |
| + | Plus operator |
| REGEXP | Uses a regular expression for matching. |
| >> | Bitwise right shift operator |
| RLIKE | Uses a regular expression for matching. It is the same as REGEXP。 |
| * | Multiplication operator |
| - | Takes the opposite value of the parameter. |
| XOR | Logical XOR |
| Coalesce | Returns the first non-NULL parameter. |
| GREATEST | Returns the largest parameter value. |
| LEAST | It returns the smallest parameter value. |
| STRCMP | Compares two strings. |

- Process control functions

| Function | Description |
| --- | --- |
| CASE | Case operator |

| Function | Description |
| --- | --- |
| IF() | If/else structure |
| IFNULL() | Null if/else structure |
| NULLIF() | If expr1 = expr2, NULL is returned. |

- Numeric functions

| Function | Description |
| --- | --- |
| ABS() | Returns the absolute value. |
| ACOS() | Returns the arc cosine of a number. |
| ASIN() | Returns the arc sine of a number. |
| ATAN2() | Returns the arc tangent of two parameters. |
| ATAN() | Returns the arc tangent of a parameter. |
| CEIL() | Obtains the smallest integer greater than or equal to a number. |
| CEILIG() | Obtains the smallest integer greater than or equal to a number. |
| CONV() | Converts a number between different number bases. |
| COS() | Returns the cosine of a number. |
| COT() | Returns the cotangent of a number. |
| CRC32() | Calculates the cyclic redundancy check (CRC) value. |
| DEGREES() | Converts a radian to a degree. |
| DIV | Returns an integer obtained from integer division. |
| EXP() | Returns e raised to the power of the specified number. |
| FLOOR() | Obtains the largest integer less than or equal to a number. |
| LN() | Returns the natural logarithm of a parameter. |
| LOG10() | Returns the logarithm with the base 10 of the parameter. |
| LOG2() | Returns the logarithm with the base 2 of the parameter. |
| LOG() | Returns the natural logarithm of the first parameter. |
| MOD() | Returns the remainder of a number. |
| %,MOD | Returns the remainder of a number divided by another number. |

| Function | Description |
| --- | --- |
| PI() | Returns the value of Pi. |
| POW() | Returns N power of the first parameter, where N is the second parameter. |
| POWER() | Returns N power of the first parameter, where N is the second parameter. |
| RADIANS() | Converts a parameter into a radian. |
| RAND() | Returns a random floating-point number. |
| ROUND() | Rounds up or down to an integer. |
| SIGN() | Returns the positive or negative sign of a parameter. |
| SIN() | Returns the sine value of a parameter. |
| SQRT() | Returns the square root of a parameter. |
| TAN() | Returns the tangent of a parameter value. |
| TRUNCATE( | Truncates to the specified decimal place. |

- String functions

| Function | Description |
| --- | --- |
| ASCII() | Returns the ASCII value of a character. |
| BIN() | Returns the binary value of a character. |
| BIT_LENGTH() | Returns the bit length of a string. |
| CHAR_LENGTH() | Returns the number of characters in a string. |
| CHAR() | Converts an input integer into a character. |
| CHARACTER_LENGTH() | Returns the number of characters in a string. It is the same as CHAR_LENGTH(). |
| CONCAT_WS() | Connects the input parameters by using the specified separator. |
| CONCAT() | Returns a connection string. |
| ELT() | Returns the string at the index number. |
| EXPORT_SET() | - |
| FIELD() | Returns the index position of the first parameter in subsequent parameters. |
| FIND_IN_SET() | Returns the index position of the first parameter in the second parameter. |

| Function | Description |
| --- | --- |
| FORMAT() | Returns the formatted numbers of the specified decimal places. |
| HEX() | It converts a decimal number or string into a hexadecimal number. |
| INSERT() | Inserts a substring of a specified number of characters at the specified place. |
| INSTR() | Returns the index position where the substring appears for the first time. |
| LCASE() | Converts to lowercase letters. It is the same as LOWER(). |
| LEFT() | Returns the characters of the specified number that is the furthest left. |
| LENGTH() | Returns the number of bytes of a string. |
| LIKE | Finds a specific character string matches a specified pattern. |
| LOCATE() | Returns the position where the substring appears for the first time. |
| LOWER() | Converts to lowercase letters. |
| LPAD() | Pads the left side of a string with a specific set of characters. |
| LTRIM() | Removes spaces at the beginning. |
| MAKE_SET() | Returns a set value (a string containing substrings separated by , characters) consisting of the characters specified in the first argument. |
| MID() | Extracts a substring from a string (starting at the specified position). |
| NOT LIKE | Finds a specific character string that does not match a specified pattern. |
| NOT REGEXP | Performs a pattern match of a string expression against a pattern. |
| OCT() | Converts a number to an octal number by a string. |
| OCTET_LENGTH() | Returns the number of bytes of a string. It is the same as LENGTH(). |
| ORD() | Returns the code for the leftmost character of the given parameter. |
| POSITION() | Returns the position where the sub-string occurs for the first time. It is the same as LOCATE(). |
| QUOTE() | Escapes parameters for use in SQL statements. |
| REPEAT() | Repeats a string for a specified number of times. |
| REPLACE() | Replaces the specified string in all the places where it appears. |

| Function | Description |
| --- | --- |
| REVERSE() | Reverses characters in a string. |
| RIGHT() | Returns the characters of the specified number that is the furthest right. |
| RPAD() | Pads strings for the specified number of times from the right. |
| RTRIM() | Removes spaces at the end. |
| SPACE() | Returns a string consisting of specified spaces. |
| STRCMP() | Compares two strings. |
| SUBSTR() | Returns the specified substring. |
| SUBSTRING_INDEX() | Returns the substring that appears for a specified number of times and in front of a separator in a string. |
| SUBSTRING() | Returns the specified substring. |
| TRIM() | Removes spaces at the beginning and end. |
| UCASE() | Converts a string to all uppercase. It is the same as UPPER(). |
| UNHEX() | Returns a string that is the hexadecimal value of the parameter. |
| UPPER() | Converts a string to uppercase. |

- Time functions

| Function | Description |
| --- | --- |
| ADDDATE() | Adds a time value (an interval) to a date. |
| ADDTIME() | Adds a time interval to a time/datetime and then returns the time/datetime. |
| CURDATE() | Returns the current date. |
| CURRENT_DATE() | Returns the current date. It is the same as CURDATE(). |
| CURRENT_TIME() | Returns the current time. It is the same as CURTIME(). |
| CURRENT_TIMESTAMP() | Returns the current date and time. It is the same as NOW(). |
| CURTIME() | Returns the current time. |
| DATE_ADD() | Adds a time value (an interval) to a date. |
| DATE_FORMAT() | Formats the date as required. |
| DATE_SUB() | Subtracts a specified time value (an interval) from a date. |

| Function | Description |
| --- | --- |
| DATE() | Extracts the date from the date expression or datetime expression. |
| DATEDIFF() | Subtracts one date from the other date. |
| DAY() | Returns the day (0-31) of a month for the specified date. It is the same as DAYOFMONTH(). |
| DAYNAME() | Returns the weekday for a date. |
| DAYOFMONTH() | Returns the day (0-31) of a month for the specified date. |
| DAYOFWEEK() | Returns the day of a week (1 for Sunday and 7 for Saturday) for a date. |
| DAYOFYEAR() | Returns the day (1-366) of a year for a date. |
| EXTRACT() | Extracts a part of a date. |
| FROM_DAYS() | Converts a day to a date. |
| FROM_UNIXTIME() | Formats a UNIX timestamp as a date. |
| GET_FORMAT() | Returns a string of the date format. |
| HOUR() | Extracts hours from input time parameters. |
| LAST_DAY() | Returns the last day of the month for the parameter. |
| LOCALTIME() | Returns the current date and time. It is the same as NOW(). |
| LOCALTIMESTAMP, LOCALTIMESTAMP() | Returns the current date and time. It is the same same as NOW(). |
| MAKEDATE() | Returns the date, containing the year and the number of days. |
| MAKETIME() | Constructs a time containing the hour, minute, and second. |
| MICROSECOND() | Returns the microsecond of a parameter. |
| MINUTE() | Returns the minute of a parameter. |
| MONTH() | Returns the month of the input date. |
| MONTHNAME() | Returns the name of a month. |
| NOW() | Returns the current date and time. |
| PERIOD_ADD() | Adds a period to a date containing the year and month. |
| PERIOD_DIFF() | Returns the number of months between two periods. |
| QUARTER() | Returns the quarter of the date parameter. |

| Function | Description |
| --- | --- |
| SEC_TO_TIME() | Converts the second into the time in 'HH:MM:SS' format. |
| SECOND() | Returns the second (0-59) of a minute. |
| STR_TO_DATE() | Converts the string to a date. |
| SUBDATE() | Subtracts a specified time value (an interval) from a date when three parameters are called. It is the same as DATE_SUB(). |
| SUBTIME() | Subtracts a time interval from a time/datetime and then returns the time/datetime. |
| SYSDATE() | Returns the function execution time. |
| TIME_FORMAT() | Formats the time. |
| TIME_TO_SEC() | Converts a parameter into a second. |
| TIME() | Extracts the time of an input parameter. |
| TIMEDIFF() | Returns the difference between two time/datetime expressions. |
| TIMESTAMP() | Returns a datetime value or expression based on a date or datetime value. If there are two arguments specified with this function, it first adds the second argument to the first, and then returns a datetime value. |
| TIMESTAMPADD() | Adds a time interval to the datetime expression. |
| TIMESTAMPDIFF() | Subtracts a time interval from the datetime expression. |
| UNIX_TIMESTAMP() | Returns the UNIX timestamp. |
| UTC_DATE() | Returns the current UTC date. |
| UTC_TIME() | Returns the current UTC time. |
| UTC_TIMESTAMP() | Returns the current UTC date and time. |
| WEEKDAY() | Returns the weekly index, where 1 indicates Sunday and 7 indicates Saturday. |
| WEEKOFYEAR() | Returns the number of the week on the calendar for a date. |
| YEAR() | It returns the year. |

- Type conversion functions

| Function | Description |
| --- | --- |
| BINARY | Converts a string into a binary string. |

| Function | Description |
| --- | --- |
| CAST() | Converts a value into a type. |
| CONVERT() | It converts a value into a type. |

# 5.4.13. Table sharding

PolarDB-X provides convenient table sharding and changing functions, allowing you to flexibly partition a table into table shards, to glue table shards to a table, and to transfer data from one table shard to another.

# 5.4.14. Multi-zone instances

PolarDB-X allows you to select a multi-zone PolarDB-X instance. This ensures the PolarDB-X instance availability when one of the zones is unavailable.

# 5.4.15. Zone-disaster recovery

PolarDB-X provides the zone-disaster recovery function, supporting migration between single-zone instances and dual-zone instances. Zone-based disaster recovery can be performed if an inappropriate zone is selected for the target PolarDB-X instance or the available ApsaraDB RDS for MySQL instances in the target PolarDB-X zone are insufficient.