Legal disclaimer

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

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

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

3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.
5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please contact Alibaba Cloud directly if you discover any errors in this document.
# Document conventions

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

Legal disclaimer........................................................................................................ I
Document conventions............................................................................................. I
1 Preface..................................................................................................................... 1
2 Limits of ApsaraDB RDS for MySQL.................................................................. 3
3 Memory management of an ApsaraDB RDS for MySQL instance...................... 5
4 Features of ApsaraDB RDS for MySQL............................................................... 9
5 Quick start.............................................................................................................. 31
  5.1 General workflow to use RDS for MySQL......................................................... 31
  5.2 Create an ApsaraDB RDS for MySQL instance................................................ 32
  5.3 Configure a whitelist for an ApsaraDB RDS for MySQL instance.................... 42
  5.4 Create databases and accounts for an ApsaraDB RDS for MySQL instance..... 54
  5.5 Connect to an RDS MySQL instance............................................................... 62
  5.6 What do I do if I cannot connect an ECS instance to an ApsaraDB for RDS instance?............................................................................................................. 67
  5.7 Instance scale-out............................................................................................. 71
    5.7.1 Billing cases for disaster recovery instances............................................. 71
    5.7.2 Disaster recovery instances..................................................................... 73
6 Data Migration....................................................................................................... 78
  6.1 Overview of data migration.............................................................................. 78
  6.2 Create a PolarDB for MySQL cluster from an ApsaraDB RDS for MySQL instance... 79
  6.3 Clone data from RDS MySQL to PolarDB MySQL with one click.................... 89
  6.4 Data Migration from a User-created Database to an ApsaraDB RDS MySQL Instance............................................................................................................... 92
    6.4.1 Migrate data from a user-created MySQL database to an ApsaraDB RDS for MySQL instance.................................................................................. 92
    6.4.2 Migrate data from a user-created Oracle database to an ApsaraDB RDS for MySQL instance................................................................. 101
    6.4.3 Migrate data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an ApsaraDB RDS for MySQL database.......................................................... 111
    6.4.4 Migrate data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an ApsaraDB RDS for MySQL instance across Alibaba Cloud accounts.......................... 121
    6.4.5 Migrate data from a user-created Db2 database to an ApsaraDB RDS for MySQL database......................................................................................... 132
    6.4.6 Migrate data to an RDS MySQL instance by using mysqldump.................. 140
  6.5 Migrate data from a third-party cloud database to ApsaraDB for RDS............. 144
6.5.1 Migrate a MySQL database from Google Cloud to Alibaba Cloud......... 144
6.5.2 Migrate data from an Amazon RDS for MySQL database to an ApsaraDB
RDS for MySQL database................................................................................ 151
6.6 Migrate data between ApsaraDB for RDS instances................................. 161
6.7 Migrate data from an ApsaraDB RDS for MySQL database to a user-created
MySQL database............................................................................................ 169

7 Data Synchronization.................................................................................... 179
7.1 Overview of data synchronization................................................................. 179
7.2 Synchronize data from MySQL to MySQL..................................................... 179
  7.2.1 Configure one-way data synchronization between ApsaraDB RDS for
    MySQL instances......................................................................................... 179
  7.2.2 Configure two-way data synchronization between ApsaraDB RDS for
    MySQL instances......................................................................................... 188
  7.2.3 Synchronize data from a user-created MySQL database connected over
    Express Connect, VPN Gateway, or Smart Access Gateway to an ApsaraDB
    RDS for MySQL instance............................................................................ 201
  7.2.4 Synchronize data between ApsaraDB RDS for MySQL instances that
    belong to different Alibaba Cloud accounts................................................ 210
  7.2.5 Synchronize data from a user-created MySQL database hosted on ECS
to an ApsaraDB RDS for MySQL instance................................................... 218
  7.2.6 Synchronize data from an ApsaraDB RDS for MySQL instance to a user-
    created MySQL database connected over Express Connect, VPN Gateway,
or Smart Access Gateway............................................................................... 227
7.3 Synchronize data from MySQL to other databases......................................... 236
  7.3.1 Synchronize data from an ApsaraDB RDS MySQL instance to a
    MaxCompute project.................................................................................... 236
  7.3.2 Synchronize data from an ApsaraDB RDS for MySQL instance to an
    AnalyticDB for MySQL cluster.................................................................... 248
  7.3.3 Synchronize data from an ApsaraDB RDS for MySQL instance to an
    AnalyticDB for PostgreSQL instance....................................................... 257

8 Manage pending events................................................................................. 265
9 Instance Lifecycle......................................................................................... 267
  9.1 Create an ApsaraDB RDS for MySQL instance............................................ 267
  9.2 Restart an RDS MySQL instance............................................................... 277
  9.3 Renew instance.......................................................................................... 278
    9.3.1 Manually renew an RDS MySQL instance............................................ 278
    9.3.2 Automatically renew an RDS MySQL instance...................................... 280
  9.4 Release an RDS MySQL instance............................................................... 284
  9.5 Manage ApsaraDB RDS for MySQL instances that are in the recycle bin..... 286

10 Instance Change.......................................................................................... 289
  10.1 Change specifications................................................................................. 289
  10.2 Manually or automatically switch over services between the RDS MySQL
    master and slave instances........................................................................... 296
  10.3 Set the maintenance window of an ApsaraDB RDS MySQL instance......... 300
10.4 Migrate an ApsaraDB RDS MySQL instance across zones ........................................ 302
10.5 Changes the data replication mode of an ApsaraDB RDS MySQL system ................. 305
10.6 Switch the billing method from pay-as-you-go to subscription ................................. 308
10.7 Switch the billing method from subscription to pay-as-you-go ............................... 309
10.8 Change the network type of an ApsaraDB RDS MySQL instance ............................ 311
10.9 Switch to a new VPC and VSwitch for an RDS MySQL instance .............................. 318

11 Version upgrade..................................................................................................... 322
11.1 Update the kernel version of an ApsaraDB RDS for MySQL instance ......................... 322
11.2 Upgrade the database engine version of an RDS MySQL instance ............................ 326

12 Instance parameters................................................................................................. 328
12.1 Reconfigure the parameters of an ApsaraDB RDS for MySQL instance .................... 328
12.2 Use a parameter template to manage parameters .................................................. 335

13 Backup..................................................................................................................... 346
13.1 Back up an ApsaraDB RDS for MySQL instance .................................................... 346
13.2 View the quota of free backup space for an ApsaraDB RDS for MySQL instance ....... 356
13.3 Download data backup files and binary log files ..................................................... 358
13.4 Back up an ApsaraDB RDS for MySQL instance across regions ............................. 362

14 Restoration.............................................................................................................. 371
14.1 Overview of data recovery ....................................................................................... 371
14.2 Restore the data of an RDS MySQL instance ......................................................... 371
14.3 Restore individual databases and tables of an ApsaraDB RDS for MySQL instance .... 377
14.4 Restore an ApsaraDB RDS for MySQL instance across regions ............................ 384
14.5 Use a physical backup file to restore an ApsaraDB RDS for MySQL instance to a user-created MySQL database ........................................................................... 390
14.6 Restore data from logical backup files of an ApsaraDB RDS MySQL instance to an on-premises database .......................................................... 397

15 Read-only instances ............................................................................................... 401
15.1 Set a replication delay for an RDS MySQL read-only instance ................................. 401
15.2 Overview of ApsaraDB RDS for MySQL read-only instances .................................. 402
15.3 Create an ApsaraDB RDS for MySQL read-only instance ...................................... 405

16 Database Autonomy Service .................................................................................... 412
16.1 DAS overview ....................................................................................................... 412
16.2 Diagnostics ............................................................................................................ 413
16.3 Autonomy center ................................................................................................... 414
16.4 Sessions Management .......................................................................................... 415
16.5 Real-time Monitoring ............................................................................................ 416
16.6 Storage Analysis .................................................................................................... 417
16.7 Capacity assessment ............................................................................................. 418
16.8 Deadlock analysis ................................................................................................ 419
16.9 Performance insight ............................................................................................. 420
16.10 Dashboard ........................................................................................................... 421
16.11 Slow Query Logs ................................................................................................ 422
17 Monitoring and alerts ................................................................. 426
17.1 View resource and engine monitoring data ................................. 426
17.2 Set the monitoring frequency of an ApsaraDB RDS for MySQL instance .......................... 428
17.3 Set Alarm Rules ........................................................................ 431

18 Database connection ................................................................. 433
18.1 Connect to an RDS MySQL instance .............................................. 433
18.2 Apply for a public endpoint for an RDS MySQL instance................. 437
18.3 Use DMS to log on to an ApsaraDB for RDS instance ................. 440
18.4 View and modify the internal and public endpoints and ports of an ApsaraDB for RDS instance ................. 444
18.5 Configure a hybrid access solution to smoothly migrate an RDS instance from the classic network to a VPC ........................................ 447

19 Database proxy ........................................................................ 454
19.1 Dedicated proxy ......................................................................... 454
19.2 Read/write splitting ...................................................................... 465
19.3 Connection pool .......................................................................... 471
19.4 Transaction splitting ..................................................................... 475
19.5 Read/write splitting ...................................................................... 477
19.5.1 Enable the read/write splitting function in the shared proxy of an ApsaraDB RDS for MySQL instance ........................................ 477
19.5.2 Change the network type of the read/write splitting address for an RDS MySQL instance ........................................ 485
19.5.3 Disable read/write splitting for an RDS MySQL instance .......... 487
19.5.4 Monitor the read/write splitting performance of an RDS MySQL instance ........................................ 488
19.5.5 Verify the read/write splitting performance of an RDS MySQL instance .......................... 489
19.5.6 Modify the latency threshold and read weights of ApsaraDB RDS for MySQL instances ........................................................................ 493
19.5.7 Rules of weight allocation by the system ......................................... 497
19.5.8 Verify the allocated read weights ................................................ 500
19.5.9 Verify the effect of read/write splitting .......................................... 501
19.5.9.1 Use SQL Explorer to verify the effect of read/write splitting ............... 501
19.5.9.2 Use internal SQL statements to verify the effect of read/write splitting ........................................ 501
19.5.10 FAQ for read/write splitting ........................................................ 503

20 Account .................................................................................. 506
20.1 Create an account for an ApsaraDB RDS for MySQL instance ........................................ 506
20.2 Reset the password of an account for an RDS MySQL instance ........................................ 511
20.3 Modify the permissions of an account for an RDS instance ........................................ 513
20.4 Reset the permissions of a premier account ........................................ 514
20.5 Authorize a service account for an RDS MySQL instance ........................................ 515
20.6 Delete an account for an RDS MySQL instance ........................................ 517
20.7 Limit permissions of a specific IP address on a database ................................ 518
20.8 Authorize accounts to manage tables, views, and fields ................................. 520
20.9 System accounts .............................................................................................. 522
20.10 Account permissions ...................................................................................... 523
21 Database ........................................................................................................... 525
   21.1 Create a database for an RDS MySQL instance ................................................. 525
   21.2 Delete a database for an RDS MySQL instance ................................................ 526
22 Data security ...................................................................................................... 529
   22.1 Switch to the enhanced whitelist mode for an RDS MySQL instance .......... 529
   22.2 Configure a whitelist for an RDS MySQL instance ........................................ 531
   22.3 Configure SSL encryption for an RDS MySQL instance ................................ 543
   22.4 Configure TDE for an ApsaraDB RDS for MySQL instance ......................... 549
   22.5 Configure disk encryption for an ApsaraDB RDS for MySQL instance .... 553
   22.6 Data security best practices in financial scenarios ........................................ 555
23 Audit .................................................................................................................. 558
   23.1 SQL Explorer ................................................................................................. 558
   23.2 Manage logs .................................................................................................. 567
   23.3 View the slow query logs of an ApsaraDB RDS for MySQL instance .......... 569
   23.4 View the event history of an ApsaraDB RDS for MySQL instance .............. 570
   23.5 Configure local binary log policies for an RDS MySQL instance ............ 575
24 Tag ...................................................................................................................... 579
   24.1 Create tags .................................................................................................. 579
   24.2 Delete tags .................................................................................................. 581
   24.3 Filter RDS instances by tag .......................................................................... 582
25 Appendixes ......................................................................................................... 584
   25.1 Commonly used SQL statements for MySQL .............................................. 584
   25.2 Grant backup file download permissions to a RAM user with only read-only
       permissions .................................................................................................... 585
   25.3 Authorize RDS to access KMS ....................................................................... 587
   25.4 Cached data persistence ............................................................................... 590
   25.5 Multi-structure data storage ......................................................................... 591
1 Preface

This topic provides an overview of RDS MySQL, including a disclaimer, terms, and concepts.

Overview

Alibaba Cloud ApsaraDB for RDS (short for Relational Database Service) is a stable, reliable, and scalable online database service. Based on Alibaba Cloud distributed file system and high-performance SSD storage, ApsaraDB for RDS supports the MySQL, SQL Server, PostgreSQL, and PPAS (compatible with Oracle) database engines and provides a portfolio of solutions to disaster tolerance, backup, recovery, monitoring, and migration to facilitate database operation and maintenance. For information about the benefits of ApsaraDB for RDS, see #unique_4.

This document describes how to configure and manage RDS through the RDS console, helping you better understand the features and functions of ApsaraDB for RDS. Additionally, you can configure and manage RDS through API and SDK.

If you need technical support, you can open the RDS console and in the upper-right corner, choose More > Support > Open a new ticket. If your business is complex, you can purchase a support plan to obtain your exclusive support service from IM enterprise groups, technical service managers (TAM), and service managers.

For more information about ApsaraDB for RDS, visit ApsaraDB RDS MySQL.

Disclaimer

Some product features or services described in this document may be unavailable for certain regions. See the relevant commercial contracts for specific Terms and Conditions.

This document serves as a user guide only. No content in this document can constitute any express or implied warranty.

Terms

- Instance: A database service process that takes up physical memory independently. You can specify the memory size, disk space, and database type of an instance, but only the memory specification determines the performance of the instance. After an instance is created, you can delete it or change its configuration as needed.

- Database: A logical unit created in an instance. Multiple databases that each have a unique name can be created in one instance.
- Region and zone: A region is a physical data center. A zone is a physical area that has independent power supply and network in a region. For more information, visit Alibaba Cloud’s Global Infrastructure.

### Concepts

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-premises database</td>
<td>A database that is deployed in your on-premises equipment room or on a cloud other than ApsaraDB for RDS.</td>
</tr>
<tr>
<td>RDS XX (XX is MySQL, SQL Server, PostgreSQL, or PPAS.)</td>
<td>A type of RDS instance. For example, RDS MySQL refers to the type of RDS instance that runs in the MySQL database engine.</td>
</tr>
</tbody>
</table>
2 Limits of ApsaraDB RDS for MySQL

To ensure instance stability and security, ApsaraDB RDS for MySQL has some service limits.

Limits of other database engines are listed as follows:

- Limits of ApsaraDB RDS for SQL Server
- Limits of ApsaraDB RDS for PostgreSQL
- Limits of ApsaraDB RDS for PPAS
- Limits of ApsaraDB RDS for MariaDB

The following tables lists the limits of ApsaraDB RDS for MySQL.

<table>
<thead>
<tr>
<th>Item</th>
<th>Limit description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance parameters</td>
<td>Instance parameters can be modified through the RDS console or API operations. Due to security and stability considerations, only specific parameters can be modified. For more information, see #unique_10.</td>
</tr>
<tr>
<td>Root permissions of databases</td>
<td>The root or system administrator permissions are not provided.</td>
</tr>
<tr>
<td>Database backup</td>
<td>• Logical backup can be performed through the command line interface (CLI) or graphical user interface (GUI).</td>
</tr>
<tr>
<td></td>
<td>• Physical backup can only be performed through the RDS console or API operations.</td>
</tr>
<tr>
<td>Database restoration</td>
<td>• Logical restoration can be performed through the CLI or GUI.</td>
</tr>
<tr>
<td></td>
<td>• Physical restoration can only be performed through the RDS console or API operations.</td>
</tr>
<tr>
<td>Item</td>
<td>Limit description</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| ApsaraDB RDS for MySQL storage engine   | Only InnoDB is supported.  
- For performance and security considerations, we recommend that you use the InnoDB storage engine.  
- The TokuDB engine is not supported. ApsaraDB RDS for MySQL has terminated its support for the TokuDB engine since August 1, 2019 because TokuDB is no longer supported by Percona and bugs may cause business losses in extreme cases. For more information about how to switch to an available engine, see #unique_11.  
- The MyISAM engine is not supported due to inherent defects that may cause data loss. If you create a MyISAM engine table, it will be automatically converted into an InnoDB engine table. For more information, see #unique_12  
- The Memory engine is not supported. If you create Memory engine tables, they are automatically converted to InnoDB engine tables. |
| Database replication                    | ApsaraDB RDS for MySQL provides a primary/secondary replication architecture (except in the Basic Edition). The secondary instance in the architecture is hidden and cannot access your applications.                                                                                                                                                                                                                           |
| Instance restart                        | Instances must be restarted through the RDS console or API operations.                                                                                                                                                                                                                                                                                                                                  |
| Network settings                        | If a MySQL 5.5 or 5.6 instance is in the classic network and the database proxy mode is enabled for it, you are not allowed to enable net.ipv4.tcp_timestamps in SNAT mode.                                                                                                                                                                                                                         |
| Storage space                           | If the storage space usage for an instance is abnormally high, the system will lock the instance to prevent data loss that may occur due to incorrect operations. You can unlock the instance by upgrading the instance specifications.                                                                                                                                                                                      |
| Help information                        | You cannot query help topic details because the mysql.help_topic table is empty.                                                                                                                                                                                                                                                                                                                      |
3 Memory management of an ApsaraDB RDS for MySQL instance

You can configure the `innodb_buffer_pool_size` parameter for an RDS for MySQL instance to improve performance.

Context
The memory of an RDS for MySQL instance is shared by the following items:

- The mysqld services. The InnoDB buffer pool of the mysqld services use memory resources. The size of the InnoDB buffer pool is specified by the `innodb_buffer_pool_size` parameter.
- The RDS management services.
- The underlying operating system.

RDS instance with Local SSDs
On an RDS for MySQL instance with local SSDs, the memory resources can all be used to run the mysqld services, including the InnoDB buffer pool and other MySQL services such as session connections.

Local SSDs are assigned from isolated storage resources on the physical server that hosts the RDS instance. The RDS management services and operating system do not use memory resources of the RDS instance.

RDS instance with standard or enhanced SSDs
On an RDS for MySQL instance with standard or enhanced SSDs, the memory resources are used to run the mysqld services, RDS management services, and underlying operating system.

The standard or enhanced SSDs are deployed on different ECS instances. Therefore, memory resources must be allocated to run the RDS management services and underlying operating system.

Generally, the underlying operating system uses 500 MB to 700 MB of memory, and the RDS management services use 500 MB of memory.
InnoDB buffer pool

You can configure the innodb_buffer_pool_size parameter for an RDS instance to specify the size of the InnoDB buffer pool. The value is configured in the following format:

{DBInstanceClassMemory*X/Y}

Example:

{DBInstanceClassMemory*7/10}

**Note:**

- DBInstanceClassMemory indicates the memory system variable of the RDS instance.
- X indicates the numerator, and Y indicates the denominator.
- The value range is $[128 \text{ MB}, DBInstanceClassMemory \times 8/10]$. It indicates that the minimum size of the InnoDB buffer pool is 128 MB and the maximum size is 80% of the RDS instance memory.

The default size of the InnoDB buffer pool for an RDS instance is as follows:

- If the RDS instance memory is smaller than 16 GB, the default size of the InnoDB buffer pool is calculated as follows: $(\text{RDS instance memory} - \text{Memory required by the RDS management services} - \text{Memory required by the underlying operating system}) \times 0.7$.
- If the RDS instance memory is larger than or equal to 16 GB or the RDS instance uses local SSDs, the default size of the InnoDB buffer pool is calculated as follows: $\text{RDS instance memory} \times 0.7$.

To ensure stability of the instance, the default value of the innodb_buffer_pool_size parameter is set to 256 MB for instances with 1 CPU core and 1 GB or 2 GB of memory.

The size of the InnoDB buffer pool must be a multiple of the calculation result of $\text{innodb_buffer_pool_chunk_size} \times \text{innodb_buffer_pool_instances}$. If you specify a size that is not a multiple of the calculation result, the system automatically changes the size. For example, if the calculation result of $\text{innodb_buffer_pool_chunk_size} \times \text{innodb_buffer_pool_instances}$ is 1 GB and you set the innodb_buffer_pool_size value to 1.5 GB, the system change the value of the innodb_buffer_pool_size parameter to 2 GB.

**Change the size of the InnoDB buffer pool for a single instance**

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID.

4. In the left-side navigation pane, click Parameters.

5. Find the `innodb_buffer_pool_size` parameter and click the icon on the right to change the value. Then, click Confirm.

   **Warning:**
   After you change the value of the `innodb_buffer_pool_size` parameter, the instance restarts. Proceed with caution.

6. In the upper-right corner of the parameter list, click Apply Changes. In the Edit Parameters dialog box, click Confirm. Then, wait for the instance to restart.

Change the size of the InnoDB buffer pool by using a parameter template

Create a parameter template and configure the value of the `innodb_buffer_pool_size` parameter. Then, apply the template to the target instance. If you have already created a parameter template, you can modify the template and then apply it. For more information, see Use a parameter template to manage parameters.
# 4 Features of ApsaraDB RDS for MySQL

This topic provides an overview of the features supported by RDS for MySQL with different database engine versions, RDS editions, and storage types.

## RDS MySQL 8.0

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
<td>Local SSD</td>
<td>Standard SSD</td>
<td>Standard SSD</td>
</tr>
<tr>
<td>Migrate the data of an RDS instance</td>
<td>Overview of data migration</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Data synchronization</td>
<td>Overview of data synchronization</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Instance management</td>
<td>Create an RDS instance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Configuration change</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Migrate an RDS instance across zones</td>
<td>✓</td>
<td>✓</td>
<td>Not supported</td>
<td>Not supported</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 8.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
<td>Local SSD</td>
<td>Standard SSD</td>
<td>ESSD</td>
</tr>
<tr>
<td></td>
<td>Manually or automatically switch over services from a primary RDS instance to its secondary instance</td>
<td>Not supported</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Change the data replication mode</td>
<td>Not supported</td>
<td>√</td>
<td>Not supported</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>Use a parameter template to manage parameters</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Create a disaster recovery instance</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Restart an RDS instance</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Set the maintenance window of an RDS instance</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 8.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
<td>Local SSD</td>
<td>Standard SSD</td>
<td>ESSD</td>
</tr>
<tr>
<td>Release an RDS instance</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Manage ApsaraDB RDS for MySQL instances that are in the recycle bin</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Instance upgrade</td>
<td>Upgrade the kernel version of an RDS instance</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Upgrade the database engine version of an RDS instance</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Not supported</td>
</tr>
<tr>
<td>Manage a database account</td>
<td>Create an account to manage databases in an RDS instance</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Reset the password of an account for an RDS instance</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 8.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>-----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
<td>Local SSD</td>
<td>Standard SSD</td>
<td>ESSD</td>
</tr>
<tr>
<td></td>
<td>Modify the permissions of an account for an RDS instance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Authorize the service account of an RDS instance</td>
<td>✓</td>
<td>✓</td>
<td>Not supported</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>Delete an account from an RDS instance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Reset the permissions of the privileged account for an RDS instance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Database management</td>
<td>Create a database for an RDS instance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Delete a database from an RDS instance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
## 4 Features of ApsaraDB RDS for MySQL

### Database connection

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
<th>MySQL 8.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
</tr>
<tr>
<td>Database connection</td>
<td>Connect to an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Configure endpoints for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>View the endpoints and ports of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Apply for a public endpoint for an RDS instance</td>
<td>✓</td>
</tr>
</tbody>
</table>

### Monitoring and alerting

<p>|                       | View the monitoring data of an RDS instance | ✓         | ✓         | ✓           | ✓    | ✓           |
|                       | Set the monitoring frequency of an RDS instance | ✓         | ✓         | ✓           | ✓    | ✓           |
|                       | Configure an alert rule for an RDS instance | ✓         | ✓         | ✓           | ✓    | ✓           |</p>
<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
<th>MySQL 8.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
</tr>
<tr>
<td>Network management</td>
<td>Change the network type of an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Switch an RDS instance to a new VPC and VSwitch</td>
<td>√</td>
</tr>
<tr>
<td>Read-only instance and read/write splitting</td>
<td>Create a read-only instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Enable read/write splitting for an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Change the network type of the read/write splitting endpoint for an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td>Security management</td>
<td>Configure a whitelist for an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 8.0</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
</tr>
<tr>
<td>Switch an RDS instance to the enhanced whitelist mode</td>
<td>Not supported</td>
<td>Not supported</td>
</tr>
<tr>
<td>Configure SSL encryption</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Configure TDE for an RDS instance</td>
<td>Not supported</td>
<td>√</td>
</tr>
<tr>
<td>Audit</td>
<td>SQL explorer</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Logs</td>
<td>√</td>
</tr>
<tr>
<td>Database backup</td>
<td>Back up the data of an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>A free tier of backup space for an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Download a data backup file or binary log file of an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 8.0</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Enterprise Edition</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Local SSD</strong></td>
</tr>
<tr>
<td>Back up the data of an RDS instance across regions</td>
<td>Not supported</td>
<td>√</td>
</tr>
<tr>
<td>Data restoration</td>
<td>Restore the data of an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Restore individual databases or tables of an RDS instance</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>Restore the data of an RDS instance across regions</td>
<td>Not supported</td>
</tr>
<tr>
<td>Dedicated proxy</td>
<td>Dedicated proxy</td>
<td>√</td>
</tr>
<tr>
<td>AliSQL</td>
<td>AliSQL</td>
<td>√</td>
</tr>
<tr>
<td>Tag management</td>
<td>Create tags</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Delete tags</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Filter RDS instances by tag</td>
<td>√</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 5.7</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
</tr>
<tr>
<td>Migrate the data of an RDS instance</td>
<td>Overview of data migration</td>
<td>√</td>
</tr>
<tr>
<td>Data synchronization</td>
<td>Overview of data synchronization</td>
<td>√</td>
</tr>
<tr>
<td>Instance management</td>
<td>Create an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Configuration change</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Migrate an RDS instance across zones</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Manually or automatically switch over services from a primary RDS instance to its secondary instance</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>Change the data replication mode</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

Issue: 20200702
## MySQL 5.7 Features of ApsaraDB RDS for MySQL

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
<th>MySQL 5.7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
</tr>
<tr>
<td>Use a parameter template to manage parameters</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Create a disaster recovery instance</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Restart an RDS instance</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Set the maintenance window of an RDS instance</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Release an RDS instance</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Manage ApsaraDB RDS for MySQL instances that are in the recycle bin</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Instance upgrade</td>
<td>Upgrade the kernel version of an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 5.7</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enterprise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
</tr>
<tr>
<td>Upgrade the database</td>
<td>Upgrade the database engine version of an RDS instance</td>
<td>Not</td>
</tr>
<tr>
<td></td>
<td></td>
<td>supported</td>
</tr>
<tr>
<td>Manage a database</td>
<td>Create an account to manage databases in an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Reset the password of an account for an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Modify the permissions of an account for an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Authorize the service account of an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delete an account from an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 5.7</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
</tr>
<tr>
<td></td>
<td>Reset the permissions of the privileged account for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td>Database management</td>
<td>Create a database for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Delete a database from an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td>Database connection</td>
<td>Connect to an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Configure endpoints for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>View the endpoints and ports of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Apply for a public endpoint for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 5.7</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Local SSD</td>
<td>Local SSD</td>
</tr>
<tr>
<td>Monitoring and alerting</td>
<td>View the monitoring data of an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Set the monitoring frequency of an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Configure an alert rule for an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td>Network management</td>
<td>Change the network type of an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Switch an RDS instance to a new VPC and VSwitch</td>
<td>√</td>
</tr>
<tr>
<td>Read-only instance and read/write splitting</td>
<td>Create a read-only instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Enable read/write splitting for an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 5.7</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
</tr>
<tr>
<td>Change the network type of the read/write splitting endpoint for an RDS instance</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Security management</td>
<td>Configure a whitelist for an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Switch an RDS instance to the enhanced whitelist mode</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Configure SSL encryption</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Configure TDE for an RDS instance</td>
<td>Not supported</td>
</tr>
<tr>
<td>Audit</td>
<td>SQL explorer</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Logs</td>
<td>√</td>
</tr>
<tr>
<td>Database backup</td>
<td>Back up the data of an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 5.7</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
</tr>
<tr>
<td>A free tier of backup space for an RDS instance</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Download a data backup file or binary log file of an RDS instance</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Back up the data of an RDS instance across regions</td>
<td>Not supported</td>
<td>√</td>
</tr>
<tr>
<td>Data restoration</td>
<td>Restore the data of an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Restore individual databases or tables of an RDS instance</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>Restore the data of an RDS instance across regions</td>
<td>Not supported</td>
</tr>
</tbody>
</table>
### ApsaraDB for RDS

#### RDS MySQL Database / 4 Features of ApsaraDB RDS for MySQL

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
<th>MySQL 5.7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
</tr>
<tr>
<td>Dedicated proxy</td>
<td>Dedicated proxy</td>
<td>√</td>
</tr>
<tr>
<td>AliSQL</td>
<td>AliSQL</td>
<td>√</td>
</tr>
<tr>
<td>Tag management</td>
<td>Create tags</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Delete tags</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Filter RDS instances by tag</td>
<td>√</td>
</tr>
</tbody>
</table>

#### RDS MySQL 5.6

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
<th>MySQL 5.6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>High-availability Edition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
</tr>
<tr>
<td>Migrate the data of an RDS instance</td>
<td>Overview of data migration</td>
<td>√</td>
</tr>
<tr>
<td>Data synchronization</td>
<td>Overview of data synchronization</td>
<td>√</td>
</tr>
<tr>
<td>Instance management</td>
<td>Create an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Configuration change</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Migrate an RDS instance across zones</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Manually or automatically switch services over from a primary RDS instance to its secondary instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Change the data replication mode of an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td></td>
<td>Apply a parameter template to an RDS instance</td>
<td>√</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 5.6</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High-availability Edition(Local SSD)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Restart an RDS instance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set the maintenance window of an RDS instance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Release an RDS instance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Manage ApsaraDB RDS for MySQL instances that are in the recycle bin</td>
</tr>
<tr>
<td></td>
<td>Instance upgrade</td>
<td>Upgrade the kernel version of an RDS instance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Upgrade the database engine version of an RDS instance</td>
</tr>
<tr>
<td></td>
<td>Manage a database account</td>
<td>Create an account to manage databases in an RDS instance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reset the password of an account for an RDS instance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Modify the permissions of an account for an RDS instance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Authorize the service account of an RDS instance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Delete an account from an RDS instance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reset the permissions of the privileged account for an RDS instance</td>
</tr>
<tr>
<td></td>
<td>Database management</td>
<td>Create a database for an RDS instance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Delete a database from an RDS instance</td>
</tr>
<tr>
<td></td>
<td>Database connection</td>
<td>Connect to an RDS instance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configure endpoints for an RDS instance</td>
</tr>
</tbody>
</table>
## Features of ApsaraDB RDS for MySQL

### MySQL 5.6

### High-availability Edition

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
<th>MySQL 5.6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Local SSD</td>
</tr>
<tr>
<td></td>
<td>View the endpoints and ports of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Apply for a public endpoint for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td>Monitoring and alerting</td>
<td>View the monitoring data of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Set the monitoring frequency of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Configure an alert rule for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td>Network management</td>
<td>Change the network type of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Switch an RDS instance to a new VPC and VSwitch</td>
<td>✓</td>
</tr>
<tr>
<td>Read-only instance and read/write splitting</td>
<td>Create a read-only instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Enable read/write splitting for a primary instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Change the network type of the read/write splitting endpoint for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td>Security management</td>
<td>Configure a whitelist for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Switch an RDS instance to the enhanced whitelist mode</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Configure SSL encryption for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Set TDE</td>
<td>✓</td>
</tr>
<tr>
<td>Audit</td>
<td>SQL explorer</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Logs</td>
<td>✓</td>
</tr>
<tr>
<td>Data backup</td>
<td>Back up the data of an RDS instance</td>
<td>✓</td>
</tr>
</tbody>
</table>
### MySQL 5.6

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
<th>MySQL 5.6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High-availability Edition</td>
<td></td>
</tr>
<tr>
<td>Local SSD</td>
<td>A free tier of backup space for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Download a data backup file or binary log file of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Back up the data of an RDS instance across regions</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Database restoration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Restore the data of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Restore individual databases or tables of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Cross-region restoration</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Dedicated proxy</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>AliSQL</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Tag management</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create tags</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Delete tags</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Filter RDS instances by tag</td>
<td>✓</td>
</tr>
</tbody>
</table>

### MySQL 5.5

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
<th>MySQL 5.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High-availability Edition</td>
<td></td>
</tr>
<tr>
<td>Local SSD</td>
<td>Migrate the data of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Data synchronization</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Instance management</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Create an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Configuration change</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Migrate an RDS instance across zones</td>
<td>✓</td>
</tr>
</tbody>
</table>
### Features of ApsaraDB RDS for MySQL

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature</th>
<th>MySQL 5.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>High-availability Edition</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Local SSD</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Manually or automatically switch services over from a primary RDS instance to its secondary instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Change the data replication mode of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Apply a parameter template to an RDS instance</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>Create a disaster recovery instance</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>Restart an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Set the maintenance window of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Release an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Manage ApsaraDB RDS for MySQL instances that are in the recycle bin</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td><strong>Instance upgrade</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Upgrade the kernel version of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Upgrade the database engine version of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td><strong>Account management</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create an account to manage databases in an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Reset the password of an account for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Modify the permissions of an account for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Authorize the service account of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Delete an account from an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 5.5</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Database management</td>
<td>Reset the permissions of the privileged account for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Create a database for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Delete a database from an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td>Database connection</td>
<td>Connect to an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Configure endpoints for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>View the endpoints and ports of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Apply for a public endpoint for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td>Monitoring and alerting</td>
<td>View the monitoring data of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Set the monitoring frequency of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Configure an alert rule for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td>Network management</td>
<td>Change the network type of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Switch an RDS instance to a new VPC and VSwitch</td>
<td>✓</td>
</tr>
<tr>
<td>Read-only instance and read/write splitting</td>
<td>Create a read-only instance</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>Enable read/write splitting for an RDS instance</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>Change the network type of the read/write splitting endpoint for an RDS instance</td>
<td>Not supported</td>
</tr>
<tr>
<td>Security management</td>
<td>Configure a whitelist for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td>Category</td>
<td>Feature</td>
<td>MySQL 5.5</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>Local SSD</td>
<td>High-availability Edition</td>
</tr>
<tr>
<td></td>
<td>Switch an RDS instance to the enhanced whitelist mode</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Configure SSL encryption for an RDS instance</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>Configure TDE for an RDS instance</td>
<td>Not supported</td>
</tr>
<tr>
<td>Audit</td>
<td>SQL explorer</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Logs</td>
<td>✓</td>
</tr>
<tr>
<td>Data backup</td>
<td>Back up the data of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>A free tier of backup space for an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Download a data backup file or binary log file of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Back up the data of an RDS instance across regions</td>
<td>Not supported</td>
</tr>
<tr>
<td>Data restoration</td>
<td>Restore the data of an RDS instance</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Restore individual databases or tables of an RDS instance</td>
<td>Not supported</td>
</tr>
<tr>
<td></td>
<td>Restore the data of an RDS instance across regions</td>
<td>Not supported</td>
</tr>
<tr>
<td>Dedicated proxy</td>
<td>Dedicated proxy</td>
<td>Not supported</td>
</tr>
<tr>
<td>AliSQL</td>
<td>AliSQL</td>
<td>x</td>
</tr>
<tr>
<td>Tag management</td>
<td>Create tags</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Delete tags</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Filter RDS instances by tag</td>
<td>✓</td>
</tr>
</tbody>
</table>
5 Quick start

5.1 General workflow to use RDS for MySQL

This topic describes the general workflow for how to create and use an RDS for MySQL instance.

For more information about how to use other engines, see the following topics:

- General workflow to use RDS for SQL Server
- General workflow to use RDS for PostgreSQL
- General workflow to use RDS for PPAS
- General workflow to use RDS for MariaDB

Quick start flowchart

If this is the first time that you use RDS for MySQL, read Limits of ApsaraDB RDS for MySQL before you purchase an RDS for MySQL instance.

The following flowchart shows the general workflow.

1. Create an ApsaraDB RDS for MySQL instance
2. Configure a whitelist for an ApsaraDB RDS for MySQL instance
3. Apply for a public endpoint for an RDS MySQL instance
4. Create databases and accounts for an ApsaraDB RDS for MySQL instance
5. Connect to an RDS MySQL instance
5.2 Create an ApsaraDB RDS for MySQL instance

This topic describes how to create an ApsaraDB RDS for MySQL instance in the ApsaraDB for RDS console. You can also call an API operation to create an ApsaraDB RDS for MySQL instance.

For information about how to create an instance in other database engines, see the following topics:

- #unique_74
- #unique_75
- #unique_76
- #unique_77

A new ApsaraDB for RDS console is available for instance creation. You can choose the new console or switch back to the original console.

Billing

For more information, see Pricing, billing items, and billing methods.

Prerequisites

You have an Alibaba Cloud account. For more information, see Sign up with Alibaba Cloud.

Precautions

- Subscription instances cannot be converted into pay-as-you-go instances.
- Pay-as-you-go instances can be converted into subscription instances. For more information, see Switch the billing method from pay-as-you-go to subscription.
- You can create up to 30 pay-as-you-go instances in your Alibaba Cloud account. To increase this quota, submit a ticket.

Create an RDS instance in the new ApsaraDB for RDS console

1. Log on to the new ApsaraDB for RDS console.

Note:

You can click Back to Old Version to switch back to the original ApsaraDB for RDS console.
2. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Billing Method** | • **Subscription**: You must pay the subscription fee when you create an instance. We recommend that you choose subscription billing for long-term use because it is more cost-effective than pay-as-you-go billing. You receive larger discounts for longer subscription periods.  
  • **Pay-As-You-Go**: A pay-as-you-go instance is charged per hour based on your actual resource usage. We recommend that you choose pay-as-you-go billing for short-term use. You can release your pay-as-you-go instance to reduce costs when you no longer need it. |
| **Region**         | The region where the RDS instance resides.  
  • You cannot change the region after you confirm your order.  
  • To maximize access speed, we recommend that you select a region in close proximity to the geographic location where your users reside.  
  • Make sure that the RDS instance is created in the same region as the ECS instance to which you want to connect. If the RDS and ECS instances reside in different regions, they cannot communicate over an internal network and therefore cannot deliver optimal performance. |
| **Database Engine**| The database engine and version that the RDS instance runs. Select the **MySQL** database engine. Supported MySQL versions are 5.5, 5.6, 5.7, and 8.0. |

**Note:**
The database engines and versions available vary based on the region you select.
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Edition</strong></td>
<td>• <strong>Basic</strong>: Your database system only consists of one instance. Computing is separated from storage to increase cost-effectiveness.</td>
</tr>
<tr>
<td></td>
<td>• <strong>High-availability</strong>: Your database system works in the classic high-availability architecture, which consists of one primary instance and one secondary instance.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Enterprise Edition</strong>: Your database system consists of one primary instance and two secondary instances, which are located in three different zones within the same region to provide finance-level reliability.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>The RDS editions available vary based on the region and database engine version you select. For more information, see <a href="https://example.com">ApsaraDB for RDS edition overview</a>.</td>
</tr>
<tr>
<td><strong>Storage Type</strong></td>
<td>• <strong>Local SSD</strong>: A local SSD is housed on the same server as the database engine. If you use local SSDs, computing is closer to data to reduce I/O latency.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Enhanced SSD</strong>: An enhanced SSD is an ultra-high performance disk provided by Alibaba Cloud based on the new generation of distributed block storage architecture. It integrates 25 Gigabit Ethernet and remote direct memory access (RDMA) technologies to reduce latency and deliver up to 1 million random input/output operations per second (IOPS).</td>
</tr>
<tr>
<td></td>
<td>• <strong>Standard SSD</strong>: A standard SSD is an elastic block storage device designed based on the distributed storage architecture. If you use standard SSDs, computing is separated from storage. For more information, see <a href="https://example.com">Storage types</a>.</td>
</tr>
<tr>
<td><strong>Zone of Primary Node</strong></td>
<td>The zone of primary node where the RDS instance resides. Each zone is an independent physical location within a region. There is no substantive difference between zones in the same region. If your database system spans multiple zones, it provides zone-level disaster recovery.</td>
</tr>
<tr>
<td></td>
<td>You only need to select a primary zone. The system automatically assigns a secondary zone to the RDS instance.</td>
</tr>
<tr>
<td><strong>Deployment Method</strong></td>
<td>You can set the instance as multi-zone or single-zone deployment.</td>
</tr>
</tbody>
</table>
### Parameter | Description
--- | ---
**Instance Type** | • **Entry-level**: belongs to the general-purpose instance family. A general-purpose instance occupies the exclusive memory and I/O resources allocated to it, but shares CPU and storage resources with the other general-purpose instances that are deployed on the same server.  
• **Enterprise-level**: belongs to the dedicated instance family. A dedicated instance occupies the exclusive CPU, memory, storage, and I/O resources allocated to it. The top configuration of the dedicated instance family is the dedicated host. A dedicated host instance occupies all the CPU, memory, storage, and I/O resources on the server where it is housed.  
  
  **Note:**  
  Each instance type supports a specific number of CPU cores, memory capacity, maximum number of connections, and maximum IOPS. For more information, see #unique_82.

**Capacity** | The storage capacity used to store data files, system files, binary log files, and transaction files. The storage capacity increases in increments of 5 GB.  
  
  **Note:**  
  The dedicated instance family supports exclusive allocations of resources. Therefore, the storage capacity of each instance type with local SSDs in this family is fixed. For more information, see #unique_82.

3. Click **Next: Instance Configuration**.
4. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Network Type**     | • **Classic Network**: a traditional type of network.  
                           • **VPC**: A virtual private cloud (VPC) is an isolated virtual network with higher security and better performance than a classic network. If you select the VPC network type, you must also specify VPC and VSwitch of Primary Node. |
<p>| <strong>Note:</strong>            | The primary RDS instance must have the same network type as the ECS instance to which you want to connect. If both their network types are VPC, you must also make sure that they reside in the same VPC. Otherwise, they cannot communicate over an internal network. |
| <strong>Storage Engine</strong>   | The default storage engine of the instance. This parameter is only available to instances that run MySQL 8.0 in the High-availability Edition equipped with local SSDs. For information about the X-Engine developed by Alibaba Cloud. |
| <strong>Note:</strong>            | X-Engine is compatible with InnoDB but performs better than InnoDB. We recommend that you select X-Engine as the default storage engine. |
| <strong>Parameter Template</strong> | The parameter template of the instance. You can select a system parameter template or create a custom parameter template. For more information, see Use a parameter template to manage parameters. |
| <strong>Note:</strong>            | This parameter is only available to instances in High-availability Edition or Basic Edition. |
| <strong>Time Zone</strong>        | The time zone of the instance.                                                                                                              |
| <strong>Note:</strong>            | This parameter is only available to instances in High-availability Edition or Basic Edition.                                                  |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Name Case Sensitivity</td>
<td>Specifies whether table names on the instance are case-sensitive. If table names in your on-premises database are case-sensitive, we recommend that you select <strong>Case-sensitive</strong>, so you can migrate data between the instance and your on-premises database with ease.</td>
</tr>
</tbody>
</table>

**Note:**
This parameter is only available to instances in High-availability Edition or Basic Edition.

5. Click **Next: Confirm Order**.

6. Confirm the settings in the **Parameters** section, specify **Purchase Plan** and **Duration**, read and select **Terms of Service**, and click **Pay Now**. You only need to specify **Duration** when you create a subscription instance.

![Parameters](image)

**Create an RDS instance in the original ApsaraDB for RDS console**

1. Log on to the **original ApsaraDB for RDS console**.

2. Select a billing method.
   - Pay-As-You-Go: A pay-as-you-go instance is charged per hour based on your actual resource usage. We recommend that you choose pay-as-you-go billing for short-term use. You can release your pay-as-you-go instance to reduce costs when you no longer need it.
   - Subscription: You must pay the subscription fee when you create an instance. We recommend that you choose subscription billing for long-term use because it is more cost-effective than pay-as-you-go billing. You receive larger discounts for longer subscription periods.
3. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Region</strong></td>
<td>The region where the RDS instance resides. You cannot change the region after you confirm your order.</td>
</tr>
<tr>
<td></td>
<td>• To maximize access speed, we recommend that you select a region in close proximity to the geographic location where your users reside.</td>
</tr>
<tr>
<td></td>
<td>• Make sure that the RDS instance is created in the same region as the ECS instance to which you want to connect. If the RDS and ECS instances reside in different regions, they cannot communicate over an internal network and therefore cannot deliver optimal performance.</td>
</tr>
<tr>
<td><strong>Resource Group</strong></td>
<td>The resource group to which the RDS instance belongs.</td>
</tr>
<tr>
<td><strong>Database Engine</strong></td>
<td>The database engine that the RDS instance runs. Select <strong>MySQL</strong>.</td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>The version of the database engine that the RDS instance runs. Supported MySQL versions are 5.5, 5.6, 5.7, and 8.0.</td>
</tr>
</tbody>
</table>

**Note:**
The database engines available vary based on the region you select.

**Note:**
The database engine versions available vary based on the region you select.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Edition** | • **Basic**: Your database system only consists of one instance. Computing is separated from storage to increase cost-effectiveness.  
• **High-availability**: Your database system works in the classic high-availability architecture, which consists of one primary instance and one secondary instance.  
• **Enterprise Edition**: Your database system consists of one primary instance and two secondary instances, which are located in three different zones within the same region to provide finance-level reliability. The Enterprise Edition is only available in four regions: China (Hangzhou), China (Shanghai), China (Shenzhen), and China (Beijing).  |
| **Storage Type** | • **Local SSD**: A local SSD is a storage device in the server node where the database engine resides. If you use local SSDs, computing is closer to data to reduce I/O latency.  
• **Standard SSD**: A standard SSD is an elastic block storage device designed based on the distributed storage architecture. If you use standard SSDs, computing is separated from storage.  
• **Enhanced SSD**: An enhanced SSD is an ultra-high performance disk provided by Alibaba Cloud based on the new generation of distributed block storage architecture. It integrates 25 Gigabit Ethernet and RDMA technologies to reduce latency and deliver up to 1 million random IOPS.  |
| **Zone** | The zone where the RDS instance resides. Each zone is an independent physical location within a region. There is no substantive difference between zones in the same region. You can deploy your primary and secondary instances in the same zone or in different zones.  
If your database system spans multiple zones, it provides zone-level disaster recovery.  |

**Note:**
The RDS editions available vary based on the database engine version you select. For more information, see [ApsaraDB for RDS edition overview](#).
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Network Type**     | • **Classic Network**: a traditional type of network.  
• **VPC**: This is the recommended network type. A VPC is an isolated virtual network with higher security and better performance than a classic network. |
| **Note:**            | The RDS instance must have the same network type as the ECS instance to which you want to connect. If the RDS and ECS instances have different network types, they cannot communicate over an internal network. |
| **CPU and Memory**   | Each instance type supports a specific number of CPU cores, memory capacity, maximum number of connections, and maximum IOPS. For more information, see #unique_82. |
|                     | ApsaraDB for RDS provides the following instance families:  
• General-purpose instance: A general-purpose instance occupies the exclusive memory and I/O resources allocated to it, but shares CPU and storage resources with the other general-purpose instances that are deployed on the same server.  
• Dedicated instance: A dedicated instance occupies the exclusive CPU, memory, storage, and I/O resources allocated to it.  
• Dedicated host: This is the top configuration of the dedicated instance family. A dedicated host instance occupies all the CPU, memory, storage, and I/O resources on the server where it is housed.  
For example, "8 Cores, 32 GB" is a general-purpose instance type, "8 Cores, 32 GB (Dedicated Instance)" is a dedicated instance type, and "30 Cores, 220 GB (Dedicated Host)" is a dedicated host instance type. |
| **Capacity**         | The storage capacity used to store data files, system files, binary log files, and transaction files. |

4. Specify Duration and Quantity, and click **Buy Now**. You only need to specify Duration when you create a subscription instance.

**Note:**

- When you create a subscription instance, you can select **Auto-renewal**, so the system automatically renews your subscription based on the duration you specify.
For example, if you create a three-month subscription instance with Auto-renewal selected, you are charged a three-month subscription fee for each automatic renewal.

• When you create a subscription instance, you can click Add to Cart to add it to the shopping cart. Later, you can click Cart to pay for the subscription instance.

5. On the Order Confirmation page, confirm the instance configuration, read and select Product Terms of Service, and click Pay Now.

What to do next

In the top navigation bar, select the region where the RDS instance you just created resides.

After the RDS instance is created, you must configure whitelists and create accounts for it. (If you want to connect to the RDS instance over the Internet, you must also apply for a public endpoint.) After you complete these operations, you can connect to the RDS instance.
FAQ

- After I purchase an RDS instance, the ApsaraDB for RDS console does not respond nor can I find the instance I just created. Why?

There are two possible reasons:

- The RDS instance you created does not reside in the region you selected.

  In the top navigation bar, select the region where the RDS instance resides. Then you can find the RDS instance you just created.

- The zone you selected cannot provide sufficient resources.

  Resources in zones are dynamically allocated. After you confirm the purchase order, the zone you selected may fail to provide sufficient resources. As a result, the instance creation fails. We recommend that you select another zone and try again. You can visit the order list to check that the fees are refunded to you.

- How do I authorize a RAM user to manage an RDS instance?

  For more information, see Use RAM to manage ApsaraDB for RDS permissions.

Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_83</td>
<td>Creates an ApsaraDB for RDS Instance.</td>
</tr>
</tbody>
</table>

5.3 Configure a whitelist for an ApsaraDB RDS for MySQL instance

This topic describes how to configure a whitelist for an ApsaraDB RDS for MySQL instance. Only entities in whitelists can access your RDS instance.

For more information about how to configure a whitelist for other database engines, see the following topics:
• Configure a whitelist for an ApsaraDB RDS for SQL Server instance
• Configure a whitelist for an ApsaraDB RDS for PostgreSQL instance
• Configure a whitelist for an ApsaraDB RDS for PPAS instance
• Configure a whitelist for an ApsaraDB RDS for MariaDB instance

Background information

ApsaraDB RDS for MySQL provides two types of whitelists:

• IP address whitelists

An IP address whitelist contains the IP addresses of entities that require access to your RDS instance. The IP address whitelist labeled default contains only the default IP address 127.0.0.1, which indicates that all entities are denied access to your RDS instance.

Before you configure an IP address whitelist, you must confirm the network isolation mode of your RDS instance. The configuration procedure varies depending on the network isolation mode used.

- Standard whitelist mode

In standard whitelist mode, an IP address whitelist can contain IP addresses from both the classic network and VPCs. The standard whitelist mode may incur security
risks. We recommend that you switch the network isolation mode from standard whitelist to enhanced whitelist.

- **Enhanced whitelist mode**

  In enhanced whitelist mode, an IP address whitelist can contain only IP addresses from the classic network or VPCs. When you create an IP address whitelist, you must specify its network type.

  ![Enhanced whitelist mode](image)

- **Security groups**

  A security group serves as a virtual firewall to limit the inbound and outbound traffic of ECS instances in that security group. After you add a security group, all ECS instances in it are granted access to your RDS instance.

  For more information, see [Create a security group](#).

  Whitelists make your RDS instance more secure, and the configuration process does not interrupt the operation of your RDS instance. Therefore, we recommend that you maintain whitelists on a regular basis.

  **Precautions for configuring an IP address whitelist**

  - You can edit or clear a default IP address whitelist, but cannot delete it.
  - You can configure up to 200 IP address whitelists for an instance.
  - Each IP address whitelist can contain up to 1,000 IP addresses or CIDR blocks. If you want to add more than 1,000 IP addresses, we recommend that you combine them into CIDR blocks such as 192.168.1.0/24.
If you attempt to log on to Data Management Service (DMS) from your RDS instance without adding your IP address to a whitelist, DMS will prompt you to add the address. By default, DMS automatically creates a whitelist that contains your IP address.

- **ali_dms_group** (IP address whitelist of DMS) and **hdm_security_ips** (IP address whitelist of HDM) are automatically created when you use the related services. Do not modify or delete the whitelists to ensure that the services run normally.

**Note:**
Do not add your business IP addresses to the whitelists. Otherwise, your business IP addresses will be overwritten during updates of the related services, failing to access the RDS instance.

## Configure an IP address whitelist in enhanced whitelist mode

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the instance resides.
3. Find the instance and click the instance ID.
4. In the left-side navigation pane, click **Data Security**.
5. Confirm the connection scenario and perform its required operations.

<table>
<thead>
<tr>
<th>Connection scenario</th>
<th>Operation</th>
</tr>
</thead>
</table>
| (Recommended) Your ECS and RDS instances reside in the same VPC. | a. On the **Whitelist Settings** tab of the Data Security page, click **Edit** to the right of the IP address whitelist labeled **default VPC**.  
    b. In the dialog box that appears, enter the internal IP address of your ECS instance in the IP Addresses field and click **OK**. |

**Note:**
Applications running on your ECS instance connect to the internal endpoint of your RDS instance.
<table>
<thead>
<tr>
<th>Connection scenario</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your ECS and RDS instances reside in different VPCs.</td>
<td><strong>a.</strong> Navigate to the <strong>Database Connection</strong> page and click <strong>Switch to Classic Network</strong>. In the dialog box that appears, click OK. <strong>b.</strong> Click <strong>Switch to VPC</strong>. In the dialog box that appears, select the VPC that hosts your ECS instance and click OK.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Your ECS and RDS instances can be switched to the same VPC only if they reside in the same region. If they reside in different regions, we recommend that you use Data Transmission Service (DTS) to migrate your RDS instance to the region where your ECS instance resides. This helps ensure service availability. For more information, see <a href="#">Migrate between RDS instances</a>.</td>
</tr>
<tr>
<td></td>
<td><strong>c.</strong> Navigate to the <strong>Whitelist Settings</strong> tab of the Data Security page, and click <strong>Edit</strong> to the right of the IP address whitelist labeled <strong>default VPC</strong>. <strong>d.</strong> In the dialog box that appears, enter the internal IP address of your ECS instance in the IP Addresses field and click <strong>OK</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Applications running on your ECS instance connect to the internal endpoint of your RDS instance.</td>
</tr>
<tr>
<td>Your ECS and RDS instances both reside in the classic network.</td>
<td><strong>a.</strong> Navigate to the <strong>Whitelist Settings</strong> tab of the Data Security page, and click <strong>Edit</strong> to the right of the IP address whitelist labeled <strong>default Classic Network</strong>. <strong>b.</strong> In the dialog box that appears, enter the internal IP address of your ECS instance in the IP Addresses field and click <strong>OK</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Applications running on your ECS instance connect to the internal endpoint of your RDS instance.</td>
</tr>
</tbody>
</table>
### Connection scenario

**Your ECS instance resides in the classic network.**

**Your RDS instance resides in a VPC.**

<table>
<thead>
<tr>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>a.</strong> Migrate your ECS instance to the VPC that hosts your RDS instance. For more information, see <a href="#">Migrate an ECS instance</a>.</td>
</tr>
<tr>
<td><strong>b.</strong> Navigate to the <strong>Whitelist Settings</strong> tab of the Data Security page, and click <strong>Edit</strong> to the right of the IP address whitelist labeled <strong>default VPC</strong>.</td>
</tr>
<tr>
<td><strong>c.</strong> In the dialog box that appears, enter the internal IP address of your ECS instance in the IP Addresses field and click <strong>OK</strong>.</td>
</tr>
</tbody>
</table>

**Note:**
Your ECS and RDS instances can be switched to the same VPC only if they reside in the same region. If they reside in different regions, we recommend that you use DTS to migrate your RDS instance to the region where your ECS instance resides. This helps ensure service availability. For more information, see [Migrate between RDS instances](#).

**a.** Navigate to the **Database Connection** page and click **Switch to VPC**. In the dialog box that appears, select the VPC that hosts your ECS instance and click OK.

**Note:**
Your ECS and RDS instances can be switched to the same VPC only if they reside in the same region. If they reside in different regions, we recommend that you use DTS to migrate your RDS instance to the region where your ECS instance resides. This helps ensure service availability. For more information, see [Migrate between RDS instances](#).

**b.** Navigate to the **Whitelist Settings** tab of the Data Security page, and click **Edit** to the right of the IP address whitelist labeled **default VPC**.

**c.** In the dialog box that appears, enter the internal IP address of your ECS instance in the IP Addresses field and click **OK**.

**Note:**
Applications running on your ECS instance connect to the internal endpoint of your RDS instance.
<table>
<thead>
<tr>
<th>Connection scenario</th>
<th>Operation</th>
</tr>
</thead>
</table>
| Your host that requires access to your RDS instance resides outside the cloud. | **a.** Navigate to the **Whitelist Settings** tab of the Data Security page, and click **Edit** to the right of the IP address whitelist labeled **default Classic Network**.  
**b.** In the dialog box that appears, enter the public IP address of your host in the IP Addresses field and click **OK**. |

**Note:**
- Applications running on your host connect to the public endpoint of your RDS instance.
- For more information about how to obtain the public IP address of your host, see #unique_88

**Note:**
- On the Whitelist Settings tab of the Data Security page, you can click **Create Whitelist**. In the Create Whitelist dialog box that appears, select **VPC** or **Classic Network/Public IP** for Network Type.
- If you enter the CIDR block 10.10.0.0/24 in the IP Addresses field, all IP addresses in the 10.10.0.X format are granted access to your RDS instance.
- If you enter more than one IP address or CIDR block, make sure that they are separated with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
If you click **Add Internal IP Addresses of ECS Instances**, IP addresses of all ECS instances created in your Alibaba Cloud account are displayed. You can select the required IP addresses to add to the whitelist.

**Network Type:**
- VPC
- Classic Network/Public IP

You currently cannot configure network isolation settings. You must enable enhanced whitelists first before configuring network isolation settings.

- **Whitelist Name**: default
- **Whitelist**: 127.0.0.1

**Add Internal IP Addresses of ECS Instances**

You can add 999 more entries.

Specified IP address: If you specify the IP address 192.168.0.1, this IP address is allowed to access the RDS instance.

Specified CIDR block: If you specify the CIDR block 192.168.0.0/24, the IP addresses ranging from 192.168.0.1 to 192.168.0.255 are allowed to access the RDS instance.

When you add multiple IP addresses or CIDR blocks, separate them by a comma (no space after the comma), for example, 192.168.0.1,192.168.0.0/24.

**How to Locate the Local IP Address**

New whitelist entries take effect in 1 minute.

**Issue:** 20200702
Configure an IP address whitelist in standard whitelist mode

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the instance resides.
3. Find the instance and click the instance ID.
4. In the left-side navigation pane, click Data Security.
5. On the Whitelist Settings tab, click Edit to the right of the IP address whitelist labeled default.

   **Note:**
   You can also click Create Whitelist to create an IP address whitelist.

6. In the Edit Whitelist dialog box, enter IP addresses or CIDR blocks in the IP Addresses field and click OK.
   - If you enter the CIDR block 10.10.10.0/24 in the IP Addresses field, all IP addresses in the 10.10.10.X format are granted access to your RDS instance.
   - If you enter more than one IP address or CIDR block, make sure that they are separated with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
   - If you click Add Internal IP Addresses of ECS Instances, IP addresses of all ECS instances created in your Alibaba Cloud account are displayed. You can select the required IP addresses to add to the whitelist.

   **Note:**
After you add IP addresses or CIDR blocks to the IP address whitelist labeled default, the system deletes the default IP address 127.0.0.1.

### Cases

- Only the default IP address 127.0.0.1 is added to a whitelist in the **Data Security > Whitelist Settings** navigation path.

The default IP address 127.0.0.1 indicates that all entities are denied access. You must add the IP addresses of entities that require access to your RDS instance to the whitelist.
• The IP address in the whitelist is set to 0.0.0.0.

Enter the CIDR block 0.0.0.0/0 instead.

**Note:**
This CIDR block 0.0.0.0/0 indicates that all entities are granted access to your RDS instance. Exercise caution when adding this CIDR block.

• IP address errors are reported when your RDS instance is in enhanced whitelist mode.

For more information, see Switch the IP whitelist mode from standard to enhanced.

- If your RDS instance resides in a VPC and is connected by using its internal endpoint, make sure that the internal IP address of your ECS instance is added to the IP address whitelist labeled **default VPC**.

- If your RDS instance resides in the classic network and is connected by using its internal endpoint, make sure that the internal IP address of your ECS instance is added to the IP address whitelist labeled **default Classic Network**.

- If your RDS instance resides in a VPC and is connected by using **ClassicLink**, make sure that the internal IP address of your ECS instance is added to the IP address whitelist labeled **default VPC**.

- If your RDS instance is connected over the Internet, make sure that the public IP address of your ECS instance is added to the IP address whitelist labeled **default Classic Network**. The IP address whitelist labeled **default VPC** cannot be used to allow access from the Internet.

• The public IP addresses added to whitelists are not real egress IP addresses.

Possible reasons are as follows:

- Public IP addresses dynamically change.

- The tool or website you use to query public IP addresses yields inaccurate results.

For more information, see #unique_88

**Precautions for configuring a security group**

• You can configure a security group only if your RDS instance runs MySQL 5.6, MySQL 5.7, or MySQL 8.0.

• Your RDS instance can have both IP address whitelists and security groups at the same time. All IP addresses in the configured whitelists and all ECS instances in the configured security group can access your RDS instance.
• You can add up to 10 security groups to an instance.
• Changes to the security group are automatically synchronized to the whitelist.
• You can only add a security group of the same network type as your RDS instance.

**Note:**
If you switch the network type of your RDS instance after you add a security group, you must add a new security group of the new network type.

**Configure a security group**

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the RDS instance resides.
3. Find the RDS instance and click the instance ID. The **Basic Information** page appears.
4. In the left-side navigation pane, click **Data Security**.
5. On the **Whitelist Settings** tab, click **Add Security Group**.

**Note:**
If a security group is followed by a VPC tag, the ECS instances in it reside in VPCs.

6. Select the security group you want to add and click **OK**.

**What to do next**

Create databases and accounts for an ApsaraDB RDS for MySQL instance

**FAQ**

• Does an IP address whitelist take effect immediately after it is configured?

An IP address whitelist takes effect approximately one minute after it is configured.

• Why do I find IP address whitelists that are not created by me?

If these whitelists contain internal IP addresses, they are probably generated by other Alibaba Cloud services such as DMS or HDM and do not call operations on your service data.
• Is my RDS instance exposed to security risks if I only enable internal network access and disable Internet access?

We recommend that you change the network type of your RDS instance to VPC. Only ECS instances in the same VPC can access your RDS instance after their IP addresses are added to the whitelists. For more information, see Change the network type of an ApsaraDB RDS for MySQL instance.

Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_90</td>
<td>Queries the IP address whitelists of an ApsaraDB RDS for MySQL instance.</td>
</tr>
<tr>
<td>#unique_91</td>
<td>Modifies an IP address whitelist of an ApsaraDB RDS for MySQL instance.</td>
</tr>
</tbody>
</table>

5.4 Create databases and accounts for an ApsaraDB RDS for MySQL instance

This topic describes how to create databases and accounts for an ApsaraDB RDS for MySQL instance.

For more information about creating databases and accounts in other database engines, see the following topics:

• #unique_92
• #unique_93
• Create a database and an account
• Create databases and accounts for an RDS PPAS instance
• Create accounts and databases for an RDS MariaDB instance

Account types

ApsaraDB RDS for MySQL supports two types of database accounts: privileged and standard. You can manage all your accounts and databases in the ApsaraDB for RDS console. For more information about the permissions that can be granted to each type of account, see Account permissions.
The type of an account cannot be changed. You can delete an account and then create a new one with the same account name. For more information, see Delete an account for an RDS MySQL instance.

<table>
<thead>
<tr>
<th>Account type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Privileged account | • You can create and manage privileged accounts by using the ApsaraDB for RDS console or API operations.  
• You can create only one privileged account per instance and then use the privileged account to manage all standard accounts and databases on that instance.  
• A privileged account enables you to manage permissions at a finer level. For example, you can grant query permissions on specific tables to standard accounts.  
• A privileged account has all permissions on all the databases of the instance on which it is created.  
• A privileged account has permissions to disconnect all standard accounts on the instance on which it is created. |
| Standard account | • You can create and manage standard accounts by using the ApsaraDB for RDS console, API operations, or SQL statements.  
• You can create more than one standard account per instance. The maximum number of standard accounts allowed varies based on the database engine kernel you use.  
• You must manually grant permissions on specific databases to standard accounts.  
• A standard account does not have permissions to create, manage, or disconnect other accounts on the instance on which it is created. |

<table>
<thead>
<tr>
<th>Account type</th>
<th>Number of databases</th>
<th>Number of tables</th>
<th>Number of accounts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privileged account</td>
<td>Unlimited</td>
<td>&lt; 200,000</td>
<td>Varies based on the kernel parameter settings of the instance</td>
</tr>
<tr>
<td>Standard account</td>
<td>500</td>
<td>&lt; 200,000</td>
<td>Varies based on the kernel parameter settings of the instance</td>
</tr>
</tbody>
</table>

**Comparison between privileged and superuser accounts**

To reduce unintended operations that may affect your business, ApsaraDB RDS for MySQL does not provide a superuser account. Only a privileged account is provided for you to manage all of the databases and standard accounts that are created on your RDS instance.
Privileged account

- The privileged account is granted the highest permissions. For more information, see Permissions of various accounts.
- The privileged account has the permissions to disconnect all standard accounts.

Superuser account

- The superuser account has the permissions to terminate the connections for all queries.
- The superuser account has the permissions to modify global variables by executing the SET statement.
- The superuser account has the permissions to execute the CHANGE MASTER and PURGE MASTER LOGS statements.
- The superuser account has the permissions to edit the files on the physical server that hosts your RDS instance.

Create a privileged account

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click Accounts.
5. On the Accounts tab, click Create Account.
6. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Account</td>
<td>Enter the username of the account. The username must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• The username must be 2 to 16 characters in length.</td>
</tr>
<tr>
<td></td>
<td>• The username must start with a lowercase letter and end with a lowercase letter or digit.</td>
</tr>
<tr>
<td></td>
<td>• The username can contain lowercase letters, digits, and underscores (_).</td>
</tr>
<tr>
<td>Account Type</td>
<td>Specify the type of the account. Select <strong>Privileged Account</strong>.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password of the account. The password must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• The password must be 8 to 32 characters in length.</td>
</tr>
<tr>
<td></td>
<td>• The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.</td>
</tr>
<tr>
<td></td>
<td>• Special characters include ! @ # $ % ^ &amp; * ( ) _ + - =</td>
</tr>
<tr>
<td>Re-enter Password</td>
<td>Enter the password of the account again.</td>
</tr>
<tr>
<td>Description</td>
<td>Enter a description that helps identify the account. The description can be up to 256 characters in length.</td>
</tr>
</tbody>
</table>

7. Click **Create**.

**Create a standard account**

1. Log on to the **ApsaraDB for RDS console**.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click **Accounts**.

5. On the Accounts tab, click **Create Account**.

6. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database Account</strong></td>
<td>Enter the username of the account. The username must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• The username must be 2 to 16 characters in length.</td>
</tr>
<tr>
<td></td>
<td>• The username must start with a lowercase letter and end with a lowercase letter or digit.</td>
</tr>
<tr>
<td></td>
<td>• The username can contain lowercase letters, digits, and underscores (_)</td>
</tr>
<tr>
<td><strong>Account Type</strong></td>
<td>Specify the type of the account. Select <strong>Standard Account</strong>.</td>
</tr>
<tr>
<td><strong>Authorized Databases</strong></td>
<td>Select one or more databases on which you want to grant permissions to the account. You can choose not to specify this parameter, because you can grant the account the permissions on specific databases after the account is created.</td>
</tr>
<tr>
<td></td>
<td><strong>a.</strong> Select one or more databases from the Unauthorized Databases list and click <strong>Add</strong> to add them to the Authorized Databases list.</td>
</tr>
<tr>
<td></td>
<td><strong>b.</strong> In the Authorized Databases list, select the <strong>Read/Write, Read-only, DDL Only, or DML Only</strong> permissions on each authorized database.</td>
</tr>
<tr>
<td></td>
<td>If you want to grant the same permissions on more than one authorized database at a time, select the authorized databases and click the button in the upper-right corner. For example, click <strong>Set All to Read/Write</strong>.</td>
</tr>
</tbody>
</table>

**Note:** The button in the upper-right corner changes as you click it. For example, after you click **Set All to Read/Write**, the button changes to **Set All to Read-only**.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>Enter the password of the account. The password must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• The password must be 8 to 32 characters in length.</td>
</tr>
<tr>
<td></td>
<td>• The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.</td>
</tr>
<tr>
<td></td>
<td>• Special characters include ! @ # $ % ^ &amp; * ( ) _ + - =</td>
</tr>
<tr>
<td>Re-enter Password</td>
<td>Enter the password of the account again.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional. Enter a description that helps identify the account. The description can be up to 256 characters in length.</td>
</tr>
</tbody>
</table>

7. Click Create.

Create a database

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click Databases.
5. Click Create Database.
6. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Database Name**           | • The name of the database must be 2 to 64 characters in length.  
• The name of the database must start with a lowercase letter and end with a lowercase letter or digit.  
• The name of the database can contain lowercase letters, digits, underscores (_), and hyphens (-).  
• The name of the database must be unique within the RDS instance. |
| **Supported Character Sets**| Select the character set used by the database. You can select utf8, gbk, latin1, or utf8mb4.  
You can also select all and then select another character set from the all drop-down list. |
| **Authorized Account**      | Select one or more accounts that require access to the database.  
You can choose not to specify this parameter, because you can bind accounts to the database after the database is created.  

**Note:**  
Only the created standard accounts are displayed. The privileged account has all the permissions on all the databases and does not need authorization on databases. |
| **Account Type**            | Select the permissions that you want to grant to the selected accounts.  
You can select Read/Write, Read-only, DDL Only, or DML Only. |
| **Description**             | Optional. Enter a description that helps identify the database. The description can be up to 256 characters in length. |

7. Click Create.

Permissions of various accounts

<table>
<thead>
<tr>
<th>Account type</th>
<th>Permission</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privileged account</td>
<td>-</td>
<td>SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATIO N CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE</td>
</tr>
<tr>
<td>Account type</td>
<td>Permission</td>
<td>Operation</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Standard account</td>
<td>Read-only</td>
<td>SELECT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INSERT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EVENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TRIGGER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTER ROUTINE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEMPORARY TABLES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTER</td>
</tr>
<tr>
<td></td>
<td>DML Only</td>
<td>SELECT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXECUTE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTER</td>
</tr>
<tr>
<td></td>
<td>DML Only</td>
<td>SELECT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXECUTE</td>
</tr>
</tbody>
</table>
FAQ

- **Can I manage accounts on read-only instances?**

  No, although all of the accounts that are created on your primary instance are replicated to its read-only instances, you cannot manage the accounts on the read-only instances. The accounts only have the permissions to read data from the read-only instances.

- **Can I manage accounts at fine-grained levels, such as the source IP address and table levels?**

  Yes, you can use commands to manage the accounts at fine-grained levels, such as the IP address and table levels, after you connect to your RDS instance.

- **Does ApsaraDB for RDS provide a superuser account such as the root user?**

  No, ApsaraDB for RDS does not provide a superuser account such as the root user. This allows you to protect your RDS instance from damages such as data loss and leakage that are caused by unintended operations.

### Related operations

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_98</td>
<td>Creates an account for an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_99</td>
<td>Creates a database for an ApsaraDB for RDS instance.</td>
</tr>
</tbody>
</table>

### 5.5 Connect to an RDS MySQL instance

This topic describes how to connect to an RDS MySQL instance. After completing the initial configuration, you can connect to your RDS instance from an ECS instance or your computer.

For information about how to connect to your RDS instance in other DB engines, see the following resources:

- #unique_100
- #unique_101
- #unique_102
- #unique_103
Background information

After you create an instance, configure a whitelist, and create an account, you can use DMS, a database client, or CLI to connect to your RDS instance. You can also set the IP address, port, and account information in applications to connect.

Use DMS to connect to an RDS instance

DMS is a graphical data management service provided by Alibaba Cloud. It can be used to manage non-relational databases and relational databases, and supports data and schema management, user authorization, security audit, data trends, data tracking, BI charts, and performance and optimization.

For more information, see Use DMS to log on to an ApsaraDB for RDS instance.

Use a database client to connect to an RDS instance

ApsaraDB RDS MySQL is fully compatible with MySQL. You can connect to an RDS instance from any general-purpose database client in the similar way you connect to a MySQL database. This section describes how to use HeidiSQL to connect to an RDS instance.

1. Start HeidiSQL.
2. In the lower-left area of the Session manager dialog box, click New.
3. Enter the information of the RDS instance to be connected. The following table describes the parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network type</td>
<td>The method of connecting to the RDS instance. Select MariaDB or MySQL (TCP/IP).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Hostname/IP</td>
<td>Enter the private or public IP address of the RDS instance.</td>
</tr>
<tr>
<td></td>
<td>• If your database client is deployed in an ECS instance that is in the same region and has the same network type as the RDS instance, you can use the private IP address of the RDS instance. For example, if the ECS and RDS instances are both in a VPC located in the China (Hangzhou) region, then you can use the private IP address of the RDS instance to create a secure, efficient connection.</td>
</tr>
<tr>
<td></td>
<td>• In the other situations, use the public IP address of the RDS instance.</td>
</tr>
<tr>
<td></td>
<td>You can obtain the private and public IP addresses of the RDS instance by completing the following steps:</td>
</tr>
<tr>
<td></td>
<td>a. Log on to the <a href="https://example.com">RDS console</a>.</td>
</tr>
<tr>
<td></td>
<td>b. In the upper-left corner of the page, select the region where the RDS instance is located.</td>
</tr>
<tr>
<td></td>
<td>c. Find the RDS instance and click its ID.</td>
</tr>
<tr>
<td></td>
<td>d. On the displayed <strong>Basic Information</strong> page, find the private and public IP addresses and their corresponding port numbers.</td>
</tr>
<tr>
<td>User</td>
<td>The username of the account that you use to access the RDS instance.</td>
</tr>
<tr>
<td>Password</td>
<td>The password of the account that you use to access the RDS instance.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Port</td>
<td>The port for the RDS instance to establish a connection. If you use the private IP address of the RDS instance to establish a connection, enter the private port number. If you use the public IP address of the RDS instance to establish a connection, enter the public port number.</td>
</tr>
</tbody>
</table>

4. Click **Open**.

If the entered information is correct, the RDS instance can be connected.
Use the CLI to connect to an RDS instance

If MySQL is installed on your server, you can use the CLI to connect to an RDS instance as follows:

```bash
mysql -h<Host name> -P<Port number> -u<Username> -p<Password> -D<RDS instance name>
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>The private or public IP address of the RDS instance. For more information, see #unique_105.</td>
<td>rm-bpsxxxxxxxxxxxx.mysql.rds.aliyuncs.com</td>
</tr>
<tr>
<td>-P</td>
<td>The port for the RDS instance to establish a connection.</td>
<td>3306</td>
</tr>
<tr>
<td></td>
<td>• If you use the private IP address of the RDS instance to establish a connection, enter the private port number.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If you use the public IP address of the RDS instance to establish a connection, enter the public port number.</td>
<td></td>
</tr>
<tr>
<td>-u</td>
<td>The username of the account that you use to access the RDS instance.</td>
<td>root</td>
</tr>
</tbody>
</table>

**Note:**

- The default port number is 3306.
- If the port used by the RDS instance to establish a connection is Port 3306, you can retain the default value.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>-p</td>
<td>The password of the account that you use to access the RDS instance.</td>
<td>password233</td>
</tr>
<tr>
<td></td>
<td>Note:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This field is optional.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If you do not enter the password in this field, the system prompts you to enter the password during subsequent operations.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If you enter the password in this field, note that no spaces are allowed between -p and the entered password.</td>
<td></td>
</tr>
<tr>
<td>-D</td>
<td>The name of the RDS instance you want to access.</td>
<td>mysql</td>
</tr>
<tr>
<td></td>
<td>Note:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• This field is optional.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• You can enter only the RDS instance name with -D removed.</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5-1: Example of connecting to an RDS instance through CLI

```
[root@123 ~]# mysql -h my-dba -p mysql_password -u root -D mysql
 Welcome to the MariaDB monitor.  Commands end with ; or \q.
 Your MySQL connection id is 116
 Server version: 5.7.25-log Source distribution
 Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
 Type 'help' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [mysql]> |
```

5.6 What do I do if I cannot connect an ECS instance to an ApsaraDB for RDS instance?

This topic describes what you can do if you cannot connect an ECS instance to an RDS instance in various situations.

If you fail to connect ECS to RDS, one common reason is that the network type of the ECS instance differs from that of the RDS instance. Another common reason is that the IP
address whitelist for the RDS instance does not contain the required IP addresses. The most common reasons and corresponding solutions are as follows:

**ECS and RDS belong to different network types**

The ECS instance runs in a VPC while the RDS instance runs in a classic network.

- Solution 1 (recommended): Switch the RDS instance from its classic network to the VPC where the ECS instance resides. For detailed steps, see #unique_107.

  **Note:**
  The RDS instance must run in the same VPC as the ECS instance after the switching so that they can communicate with each other through the intranet.

- Solution 2: Purchase another ECS instance that runs in the classic network because ECS instances cannot be switched from a VPC to the classic network. A VPC is safer than the classic network. Therefore, we recommend that you use a VPC.

- Solution 3: Connect the ECS instance to the RDS instance through the Internet by using the public address of the RDS instance. This solution is inferior to solutions 1 and 2 in terms of performance, security, and stability.

The ECS instance runs in the classic network while the RDS instance runs in a VPC.

- Solution 1 (recommended): Switch the ECS instance from the classic network to the VPC where the RDS instance resides.

  **Note:**
  The ECS instance must run in the same VPC as the RDS instance after the switching so that they can communicate with each other through the intranet.

- Solution 2: Switch the RDS instance from its VPC to the classic network. However, a VPC is safer than the classic network. Therefore, we recommend that you use a VPC.

- Solution 3: Use the **ClassicLink** function. This function allows the ECS instances in the classic network to communicate with the resources in a VPC through the intranet.

- Solution 4: Connect the ECS instance to the RDS instance through the Internet by using the public address of the RDS instance. This solution is inferior to solutions 1, 2, and 3 in terms of performance, security, and stability.
ECS and RDS are in different VPCs

Each VPC is a logically isolated network on Alibaba Cloud. If the ECS instance and RDS instance both run in VPCs, they must be in the same VPC so that they can communicate with each other through the intranet.

- Solution 1 (recommended): Switch the RDS instance to the VPC where the ECS instance is located.
  Specifically, switch the RDS instance from its VPC to the classic network and then switch from the classic network to the VPC where the ECS instance resides. For detailed steps, see #unique_107.
- Solution 2: Establish an Express Connect channel between the two VPCs. For detailed steps, see Alibaba Cloud CEN tutorials.
- Solution 3: Connect the ECS instance to the RDS instance through the Internet. This solution is inferior to solutions 1 and 2 in terms of performance, security, and stability.

ECS and RDS are in different regions

If the ECS instance is located in a region different from the RDS instance, they cannot communicate with each other through the intranet.

- Solution 1: Release the ECS or RDS instance and purchase instances again.
- Solution 2: Set the network types of the ECS instance and RDS instance to VPCs, and establish an Express Connect channel between the two VPCs. For detailed steps, see #unique_107 and Alibaba Cloud CEN tutorials.
- Solution 3: Connect the ECS instance to the RDS instance through the Internet. This solution is inferior to solutions 1 and 2 in terms of performance, security, and stability.

Incorrect IP address whitelist settings

- The whitelist contains only the default IP address 127.0.0.1, which indicates that no devices are allowed to access the RDS instance. You need to add the IP address of the ECS instance to the whitelist. For detailed steps, see #unique_108.
- The IP address in the whitelist is 0.0.0.0. However, the correct format is 0.0.0.0/0.

**Note:**

0.0.0.0/0 indicates that all devices are allowed to access the RDS instance. Please use it with caution.
• The whitelist is set to the enhanced security mode. In this case, you need to check the following:
  - If you want the ECS instance to connect to the RDS instance through the VPC address, ensure that the private IP address of the ECS instance is added to the VPC whitelist of the RDS instance.
  - If you want the ECS instance to connect to the RDS instance through the classic network address, ensure that the private IP address of the ECS instance is added to the classic network whitelist of the RDS instance.
  - If you want the ECS instance to connect to the RDS instance through the Internet address, ensure that the public IP address of the ECS instance is added to the classic network whitelist of the RDS instance. The VPC whitelist does not restrict access from the Internet.
• The public IP address that you add to the whitelist is not the real outbound IP address of the ECS instance. Possible reasons are as follows:
  - The public IP address is not fixed and may change.
  - The IP address query tool or website may provide inaccurate IP addresses.

To find out the real IP address, see Locate the real IP address.

**Domain name resolution failures**

If your Domain Name Server (DNS) fails or its network interface card (NIC) configuration is changed, domain name resolution may fail. You can run the `ping` and `telnet` commands to check whether you can properly connect to the RDS instance.

```
ping <domain name>
telnet <domain name><port number>
```

**Example:**

If the communication is abnormal, you can modify the NIC configuration file of your DNS to resolve the problem by completing the following steps:

1. Modify the NIC configuration file.

```
vi /etc/sysconfig/network-scripts/<name of the NIC configuration file>
```

**Note:**
Fill the name of the NIC used by the ECS server in the <name of the NIC configuration file> field. You can run the ifconfig command to check the suffix. The default suffix is ifcfg-eth0.

2. Add the following information to the end of the NIC configuration file:

```plaintext
DNS1=100.100.2.136
DNS2=100.100.2.138
```

**Note:**
If the **DNS1** and **DNS2** parameters are set, you need to change their settings to the IP addresses shown above.

3. Run the following command to restart your network service:

```plaintext
systemctl restart network
```

4. Run the following command to check whether the modification is successful:

```plaintext
cat /etc/resolv.conf
```

### 5.7 Instance scale-out

#### 5.7.1 Billing cases for disaster recovery instances

This topic describes how to calculate fees incurred by disaster recovery instances.

**Billing case**

You use an ApsaraDB RDS for MySQL 5.6 High-availability Edition instance and select the subscription billing method. The following table describes the billing items.

**Note:**
The prices provided in this topic are for reference only. If you want to know further details about the actual prices, go to the ApsaraDB for RDS console.
<table>
<thead>
<tr>
<th>Billing item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>The subscription fee incurred for an ApsaraDB for RDS instance. The subscription fee for a year is USD 2,400.</td>
</tr>
<tr>
<td>Storage capacity</td>
<td>You are charged for the storage space of the RDS instance. The billing method is subscription. It is the same as the billing method of the instance. The price is USD 0.13/GB/month. The total cost of 2,000 GB of storage space is calculated as 2,000 x 0.13 x 12=USD 3,120.</td>
</tr>
</tbody>
</table>

If you do not select other billing items, the total cost is calculated as 2,400+3,120=USD 5,520 per year.

In this scenario, the business has high requirements for data reliability. Geo-disaster recovery instances must be deployed to enhance data reliability. We recommend that you use the disaster recovery instance of ApsaraDB for RDS. This can be directly created based on the original ApsaraDB RDS for MySQL instance.

**Note:**
For more information about how to create a disaster recovery instance, see [Disaster recovery instances](#).

After you purchase a disaster recovery instance, you must pay for the billing items described in the following table.

<table>
<thead>
<tr>
<th>Billing item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disaster recovery instance</td>
<td>The pay-as-you-go fee cost by an ApsaraDB for RDS disaster recovery instance. The specifications are the same as those for the primary instance. The pay-as-you-go price is USD 0.35/hour. The cost of a year is 0.35 x 24 x 365≈USD 3,000.</td>
</tr>
<tr>
<td>Storage capacity</td>
<td>You are charged for the storage space of the disaster recovery instance. The billing method is pay-as-you-go. It is the same as the billing method of the disaster recovery instance. The price is USD 0.0003/GB/hour. The total cost of 2,000 GB of storage space is calculated as 2,000 x 0.0003 x 24 x 365≈USD 5,200.</td>
</tr>
</tbody>
</table>

**Note:**
The prices provided in this topic are for reference only. If you want to know further details about the actual prices, go to the ApsaraDB for RDS console.
<table>
<thead>
<tr>
<th>Billing item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTS synchronization</td>
<td>The data of the primary instance is synchronized to the disaster recovery instance through Data Transmission Service (DTS). Therefore, you are charged based on the cost of DTS synchronization. The billing method is pay-as-you-go. The price is USD 0.349/hour. The cost of a year is 0.349 x 24 x 365≈USD3,000.</td>
</tr>
</tbody>
</table>

**Note:**
On the **Basic Information** page of the disaster recovery instance, you can click **View Synchronous Job Details** in the upper-left corner of the Status section to go to the DTS console and view the synchronization task details.

When you use a disaster recovery instance, the total cost of a year is calculated as 3,000+5,200+3,000=USD 11,200. Compared with user-created disaster recovery instances, ApsaraDB for RDS disaster recovery instances provide high stability and low cost for hardware and network traffic.

### 5.7.2 Disaster recovery instances

For services that require high data reliability or financial services that require regulation, RDS provides remote disaster recovery instances to improve data reliability.

**Background introduction**

RDS achieves real-time synchronization between the master instance and the remote disaster recovery instance through Data Transmission Service (DTS). Both instances are deployed in the master/slave high-availability architecture. If the master node and slave node cannot be connected due to any abrupt event such as natural disasters in the region of the master instance, you can switch the remote disaster recovery instance to the master instance. After the database link address is modified on the application, service access to the application can be quickly recovered.

Through the DTS console, disaster recovery instances can enable synchronization links to support original features such as synchronization object changing, synchronization rate setting, and delay alarming. For details, see the DTS product documentation.

Disaster recovery instances have the following features:
• Provide independent database connection addresses so that the connections can be independently controlled by user applications.
• Use the master/slave high-availability architecture.
• Support hourly billing and can be enabled and disabled out-of-the-box.
• Support independent whitelist configuration and account management.

Billing

The configuration of an RDS disaster recovery instance is exactly the same as its master instance configuration, data transmission realizes real-time synchronization between the master instance and the remote disaster recovery instance. As a result, creating disaster recovery instances will result in fees of both RDS and DTS. For price details, see ApsaraDB RDS MySQL Pricing and Data Transmission Service Pricing.

Prerequisites

• Currently, disaster recovery instances support RDS MySQL instances only.
• When a disaster recovery instance is to be created, the master instance version must be MySQL 5.6 or later. Make a compatibility test before upgrading the master instance version. Alternatively, create a new MySQL 5.6 to copy the data from the master instance to the new instance, then create a disaster recovery instance on the new instance.
• The master instance that creates the disaster recovery instance must have an intranet IP address.

Limits

Disaster recovery instances do not support the following functions: backup settings, backup recovery, data migration, database management, Internet IP address assignment, and connection address modification.

Operation steps

1. Log on to the RDS console.
2. In the upper-left corner, select the region where the target instance is located.

3. Find the target instance and click its ID.

4. On the Basic Information page, click Add Guard.

5. On the Create Synchronization Task page, click Purchase Instance Right Now to purchase a disaster recovery instance.

Parameter description:

- **Synchronization Task Name**: indicates the name of a synchronization task. The default name can be retained.
- **Instance ID** (local instance information): indicates the ID of the local instance. The system automatically associates the current instance ID, or you can click an RDS instance under a different Alibaba Cloud account. Fill in an RDS instance ID, database, account number, and the corresponding password.
- **Instance ID** (target instance information): indicates the ID of the target instance. Click Buy Now to purchase the disaster recovery instance.

6. In the target RDS instance purchase window, select the region where the instance is located, and click Buy Now.

During the purchase of a disaster recovery instance, you can only select a region, and other configuration information is consistent with that of the master instance by default. If there are any disaster recovery instance upgrade requirements, you can create...
them successfully in RDS Management Console to change configuration of the disaster recovery instance.

Note:
It takes minutes to create a disaster recovery instance. Do not close the dialog box during creation. Otherwise, the disaster recovery instance may fail to be created.

7. After the disaster recovery instance is purchased, the instance ID is automatically added to the target instance ID. Then, click Authorize White List and Next.

8. The system automatically creates a migration account. After the creation is complete, click Next.

Note:
The name is automatically generated for the disaster recovery instance. For DTS synchronization, do not modify or delete the account. Otherwise, a synchronization exception may occur.

9. Select the objects to be migrated from the source database objects, click > to add the selected objects, and then click Next.

10. Select the synchronization initialization type and set the synchronization rate, and click Pre-check to start the synchronization.

Parameter description:

- Synchronization Initialization: Migrate the structure and data of the synchronization objects from the local instance to the disaster recovery instance, which will be used as the basis for subsequent incremental data synchronization. This parameter has two options: Structure Initialization and Full Data Initialization. Both options must be selected for your first data synchronization.

- Synchronization Rate: Set the rate of synchronization between the master instance and the disaster recovery instance to protect services on the master instance. The unit is TPS. If this default value of the parameter is retained, the synchronization rate is the upper limit of the DTS performance.

Note:
Pre-check failure is described below. If pre-check is passed, go to Step 12.

11. The system displays the pre-check results.
12. Click the detection item after the test results is failed. View the failure details to complete the error troubleshooting.

13. After troubleshooting, select the current synchronization task from **Synchronization Task List** and click **Start**.

14. After the pre-check is passed, click **OK**. The synchronization task is automatically started.

15. On the data synchronization list page of DTS, you can query created synchronization tasks and operate such tasks by changing the synchronization object, setting the monitoring alarm, and modifying the synchronization rate. For details, see DTS product documentation.

**Note:**
In order to ensure the real-time performance of disaster backup instance data, do not pause the synchronization task of the disaster recovery instance.
# 6 Data Migration

## 6.1 Overview of data migration

This topic provides an overview of the data migration solutions supported by ApsaraDB RDS for MySQL.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Reference</th>
</tr>
</thead>
</table>
| Migrate data from an ApsaraDB RDS for MySQL instance to a PolarDB cluster | • Create a PolarDB for MySQL cluster from an ApsaraDB RDS for MySQL instance  
• Clone data from RDS MySQL to PolarDB MySQL with one click |
| Migrate data from a user-created database to an ApsaraDB RDS for MySQL instance | • Migrate data from a user-created MySQL database to an ApsaraDB RDS for MySQL instance  
• Migrate data from a user-created Oracle database to an ApsaraDB RDS for MySQL instance  
• Migrate data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an ApsaraDB RDS for MySQL database  
• Migrate data from a user-created Db2 database to an ApsaraDB RDS for MySQL database  
• Migrate data to an RDS MySQL instance by using mysqldump |
| Migrate data from a database on a third-party cloud platform to an ApsaraDB RDS for MySQL instance | • Migrate data from an Amazon RDS for MySQL database to an ApsaraDB RDS for MySQL database  
• #unique_120  
• #unique_121  
• Migrate a MySQL database from Google Cloud to Alibaba Cloud |
| Migrate data from an ApsaraDB RDS for MySQL instance to a user-created MySQL database | Migrate data from an ApsaraDB RDS for MySQL database to a user-created MySQL database |
6.2 Create a PolarDB for MySQL cluster from an ApsaraDB RDS for MySQL instance

You can create a PolarDB for MySQL cluster from an existing ApsaraDB RDS for MySQL instance with data migration. This topic describes how to create a PolarDB for MySQL cluster from an ApsaraDB RDS for MySQL instance.

Prerequisites

- The source RDS instance is the ApsaraDB RDS for MySQL 5.6 high-availability edition.
- Make sure that Transparent Data Encryption (TDE) and Secure Sockets Layer (SSL) are disabled on the source RDS instance.
- Make sure that the source RDS instance uses InnoDB as the table storage engine.
- If the RDS instance is running in proxy mode (safe mode), make sure that you have created a privileged account (see Create a privileged account) or switched to the high-performance mode (see #unique_124).

Context

Apsara PolarDB is the next-generation relational cloud database developed by Alibaba Cloud. It has the following benefits:

- Large storage capacity: up to 100 TB.
- High performance: up to six times higher than MySQL.
- Serverless storage: you do not need to purchase storage in advance. The storage can be automatically expanded and is charged based on usage.
- Temporary upgrade: supports temporary specification upgrade to cope with workload spikes.

For more information, see #unique_125.

After a PolarDB for MySQL cluster is created from an existing ApsaraDB RDS for MySQL instance, the Apsara PolarDB cluster contains the accounts, databases, IP whitelists, and required parameters of the source RDS instance.
Features

- Data migration is free of charge.
- No data loss during data migration.
- Supports incremental data migration. This allows you to migrate data with a service
downtime of less than 10 minutes.
- Supports migration rollback. If a migration fails, it can be rolled back within 10 minutes.
- Supports switchover with endpoint interchange. You do not need to change the
configurations of your applications to reconnect to Apsara PolarDB.

Procedure

1. Create an Apsara PolarDB cluster to store the same data as the source RDS instance.
Incremental data is then dynamically synchronized from the source RDS instance to the
Apsara PolarDB cluster. For more information, see Migrate data from the source RDS
instance.

   Note:
   You must change the database endpoint in your applications to the endpoint of the
   Apsara PolarDB cluster, verify that your workloads are running properly, and click
   Complete Migration within seven days. After you click Complete Migration, the system
   stops synchronizing data between the source RDS instance and the Apsara PolarDB
   cluster.

2. Click Switch. The operation reverses the migration to synchronize incremental data from
the Apsara PolarDB cluster to the RDS instance. After the synchronization is complete,
the source RDS instance is switched to the read-only mode and the Apsara PolarDB
cluster is switched to the read/write mode. Modify the database endpoint in your
applications at the earliest opportunity. For more information, see Perform a reverse
migration.

   Note:
   After the reverse migration is complete, you can also roll back the migration.

3. Complete the migration.

Limits

- The source RDS instance and the Apsara PolarDB cluster must be deployed in the same
region.
The parameters of the source RDS instance cannot be modified during migration.

**Migrate data from the source RDS instance.**

This operation creates an Apsara PolarDB cluster that stores the same data as the source RDS instance. Then, incremental data is dynamically synchronized from the source RDS instance to the Apsara PolarDB cluster.

1. Log on to the Apsara PolarDB console.
2. Click **Create Cluster**.
3. Select **Subscription** or **Pay-as-you-go**.
4. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region</td>
<td>Select the region where the Apsara RDS for MySQL instance is deployed.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The new Apsara PolarDB cluster is also deployed in this region.</td>
</tr>
<tr>
<td>Create Type</td>
<td>Select <strong>Migration from RDS</strong>. Apsara PolarDB uses this method to clone a</td>
</tr>
<tr>
<td></td>
<td>cluster from the source RDS instance. It keeps data synchronized between the</td>
</tr>
<tr>
<td></td>
<td>RDS instance and the Apsara PolarDB cluster. By default, the binary log of</td>
</tr>
<tr>
<td></td>
<td>the new cluster is enabled.</td>
</tr>
<tr>
<td>RDS Engine Type</td>
<td>The engine type of the source RDS instance, which cannot be modified.</td>
</tr>
<tr>
<td>RDS Engine Version</td>
<td>The engine version of the source RDS instance, which cannot be modified.</td>
</tr>
<tr>
<td>Source RDS instance</td>
<td>Select an available RDS instance from the drop-down list. Read-only RDS</td>
</tr>
<tr>
<td></td>
<td>instances are not listed.</td>
</tr>
<tr>
<td>Primary Availability Zone</td>
<td>The zone of the cluster. Each zone is an independent geographic location</td>
</tr>
<tr>
<td></td>
<td>within a region. The zones that are deployed in the same region are similar.</td>
</tr>
<tr>
<td></td>
<td>You can deploy your Apsara PolarDB cluster and the ECS instance in the same</td>
</tr>
<tr>
<td></td>
<td>zone or in different zones.</td>
</tr>
<tr>
<td>Network Type</td>
<td>The network type of the Apsara PolarDB cluster that cannot be modified.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VPC</strong></td>
<td>The VPC network and VSwitch to which the Apsara PolarDB cluster is connected. Make sure that the Apsara PolarDB cluster and ECS instance are deployed in the same VPC network. Otherwise, the cluster and the ECS instance cannot communicate with each other over the internal network and achieve optimal performance.</td>
</tr>
<tr>
<td><strong>VSwitch</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Compatibility</strong></td>
<td>The database engine of the Apsara PolarDB cluster that which cannot be modified.</td>
</tr>
<tr>
<td><strong>Node Specification</strong></td>
<td>Select a node specification for the Apsara PolarDB cluster based on your workload requirements. We recommend that you select a specification that is the same or higher than that of the source RDS instance. All nodes in the Apsara PolarDB cluster are dedicated nodes with stable performance. For more information, see #unique_130.</td>
</tr>
<tr>
<td><strong>Nodes</strong></td>
<td>The number of nodes in the new cluster. Keep the default setting. The system automatically creates a read-only node with the same specification as that of the primary node.</td>
</tr>
<tr>
<td><strong>Storage Cost</strong></td>
<td>You do not need to specify this parameter. Storage fees are charged based on the usage on an hourly basis. For more information, see #unique_130.</td>
</tr>
</tbody>
</table>

5. Specify the **Purchase Plan** parameter if you are creating a subscription cluster, and then click **Buy Now** on the right side of the page.

6. Confirm the order information, read the **Agreement of Service**, select the check box, and then click **Pay**.

7. Log on to the Apsara PolarDB console and check the status of the new Apsara PolarDB cluster.

### Note:

- After the cluster is created, it synchronizes data from the source RDS instance. You must modify the database endpoint in your applications and click **Complete Migration** within seven days. The data migration task is automatically terminated after seven days.
- You can also cancel the migration. For more information about the impacts of cancelling a migration task, see **FAQ**.
Perform a reverse migration

You can perform a reverse migration to synchronize data from the Apsara PolarDB cluster to the RDS instance if the following conditions are met, and then change the database endpoint in your applications.

- You have completed tasks described in Migrate data from the source RDS instance.
- The value of Replication Latency must be less than 60 seconds.

1. Log on to the Apsara PolarDB console.
2. Find the target cluster and click the cluster ID.
3. On the Overview page, click Switch. In the dialog box that appears, click OK. The operation reverses the migration to synchronize incremental data from the Apsara PolarDB cluster to the RDS instance. After the synchronization is completed, the source RDS instance is switched to the read-only mode and the Apsara PolarDB cluster is switched to the read/write mode.

Note:

- You cannot perform a reverse migration if the replication latency is higher than 60 seconds.
- In most cases, the reverse migration process lasts less than 5 minutes.
4. In the **Start Switching** dialog box, select **Switch with Endpoints Interchange (Connection Changes Not Required)** or **Switch with Endpoints Interchange (Connection Changes Required)**.

If you do not want to change the connection configurations of your applications, select **Switch with Endpoints Interchange (Connection Changes Not Required)**. After you select **Switch with Endpoints Interchange**, refer to the correspondence between the

<table>
<thead>
<tr>
<th>Primary Endpoint</th>
<th>RDS</th>
<th>Apsara PolarDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read-write Splitting/Cluster Endpoint</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This operation interchanges the read/write modes of the RDS for MySQL instance and Apsara PolarDB cluster, and changes the replication direction.
endpoint of the RDS instance and the endpoint of Apsara PolarDB cluster, as shown in the following figure.

**Notice:**

- If you select Switch with Endpoints Interchange, only the domain names of ApsaraDB for RDS and Apsara PolarDB are interchanged. Other configurations such as the VSwitches and VIPs are not interchanged.
- If you select Switch with Endpoints Interchange, the port numbers are not interchanged. Make sure that the port number of the ApsaraDB for RDS instance is the same as that of the Apsara PolarDB cluster. For more information, see View and modify the internal and public endpoints and ports of an ApsaraDB for RDS instance.
- After domain names are interchanged, a DNS cache issue may occur. You may fail to connect to the database before the DNS cache expires. To resolve this issue, we recommend that you refresh the DNS cache.
- If you want to use Data Management Service (DMS) to log on to the Apsara PolarDB database, you must use the latest version of DMS and log on to the database with the cluster ID.

5. Click **OK**.
6. Refresh the page. After **PolarDB Read/Write Status** changes to **Read and Write**, change the database endpoint in your applications at the earliest opportunity.

Note:
During the reverse migration, you can also roll back the migration.

**Complete migration**

After you complete the tasks described in the **Migrate data from the source RDS instance**. section, you must change the database endpoint in your applications, and click **Complete Migration** within seven days. This operation stops data synchronization between the RDS instance and the Apsara PolarDB cluster.

**Warning:**
This operation stops data synchronization between the RDS instance and the Apsara PolarDB cluster, and the rollback feature becomes unavailable. We recommend that you use the Apsara PolarDB cluster for a period of time, and click Complete Migration after you confirm that the cluster is running properly.

1. Log on to the **Apsara PolarDB console**.
2. Find the target cluster and click the cluster ID.
3. On the **Overview** page, click **Complete Migration**. In the dialog box that appears, click **OK**.

![Complete Migration dialog box](image)

**Note:**

- After you click **OK**, the system stops data synchronization within two minutes. During this process, the **Complete Migration** button is still available. Do not click it again.
- You can choose whether to disable the binary log feature for the Apsara PolarDB cluster. If this feature is disabled, the write performance can be slightly improved. However, you must restart the Apsara PolarDB cluster.

4. Release the source RDS instance if it is no longer needed.

**Roll back the migration**

Before a migration is complete, you can also roll back the migration. After the rollback operation is complete, the source RDS instance is switched to the read/write mode, and the Apsara PolarDB cluster is switched to the read-only mode. The system synchronizes data from the source RDS instance to the Apsara PolarDB cluster.

1. Log on to the **Apsara PolarDB console**.
2. Find the target cluster and click the cluster ID.

3. On the **Overview** page, click **Rollback**. In the dialog box that appears, click **OK**.

![Rollback dialog box](image)

**Note:**

After you click **OK**, the system synchronizes data from the source RDS instance to the Apsara PolarDB cluster. After the rollback operation is complete, the source RDS instance is switched to the read/write mode and the Apsara PolarDB cluster is switched to the read-only mode. After **Source RDS Read/Write Status** is displayed as **Read and Write**, change the database endpoint in your applications to the endpoint of the RDS instance at the earliest opportunity.

**FAQ**

- **Is the source RDS instance affected when data is migrated from the RDS instance?**
  
  No, the source RDS instance can run properly.

- **Does smooth migration affect my workloads running on the connected databases?**
  
  Smooth migration ensures zero data loss during migration. The service downtime is less than 10 minutes. You can roll back the migration if necessary.

- **What happens if I cancel the migration?**

  After the migration is canceled, you can modify the parameters of the source RDS instance. The Apsara PolarDB cluster is switched to the read/write mode, and the data is not released. If you manually cancel the migration, you can choose whether to disable the binary log feature for the Apsara PolarDB cluster. The binary log feature is not disabled if the migration is automatically canceled.
Related API operations

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_132</td>
<td>Creates a PolarDB cluster.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> To create a cluster by cloning an existing RDS instance with data migration, set the <strong>CreationOption</strong> parameter to <strong>MigrationFromRDS</strong>.</td>
</tr>
<tr>
<td>#unique_133</td>
<td>Queries the status of data migration for an Apsara PolarDB cluster.</td>
</tr>
<tr>
<td>#unique_134</td>
<td>Performs a reverse migration to synchronize cluster data to the source RDS instance, or rolls back the migration.</td>
</tr>
<tr>
<td>#unique_135</td>
<td>Cancels or completes the migration for an Apsara PolarDB cluster.</td>
</tr>
</tbody>
</table>

6.3 Clone data from RDS MySQL to PolarDB MySQL with one click

ApsaraDB for PolarDB allows you to clone data from an RDS MySQL instance to a new PolarDB MySQL cluster with one click.

This feature creates a destination ApsaraDB for PolarDB cluster with the same data as that of the source RDS instance. The incremental data of the source RDS instance will not be synchronized to the destination ApsaraDB for PolarDB cluster.

**Note:**
If you need to synchronize the incremental data of the source RDS instance to the destination ApsaraDB for PolarDB cluster in real time while the cluster is being created, that is, to smoothly migrate data without service interruption, see #unique_136.

ApsaraDB for PolarDB introduction

ApsaraDB for PolarDB is the next-generation relational cloud database developed by Alibaba Cloud, which has the following main advantages.

- Large storage capacity: up to 100 TB of storage.
- High performance: up to 6x performance improvement over MySQL.
• Serverless storage: no need to purchase storage capacity in advance, which is automatically scaled and is billed by usage.
• Temporary upgrade: supports temporary upgrade of specifications to easily cope with short-term business peaks.

For more information, see #unique_125.

Highlights
• Free-of-charge
• Zero data loss during cloning

Precautions
• Data cloning can only be performed in the same region.
• The destination ApsaraDB for PolarDB cluster must contain information of the source RDS instance, including the account, database, IP address whitelist, and required parameters.

Prerequisites
• The source RDS instance is of the RDS MySQL 5.6 high-availability version.
• Transparent Data Encryption (TDE) and Secure Sockets Layer (SSL) are not enabled in the source RDS instance.
• The table storage engine of the source RDS instance is InnoDB.

Procedure
1. Log on to the ApsaraDB for PolarDB console.
2. Click Create Cluster.
3. Select Subscription or Pay-As-You-Go (Hourly Rate).
4. Set parameters listed in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Region    | The region where the source RDS MySQL instance resides.  

**Note:**  
The destination ApsaraDB for PolarDB cluster is also located in this region.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Type</td>
<td>The method of creating the cluster.</td>
</tr>
<tr>
<td></td>
<td>• Default Create Type: creates a new ApsaraDB for PolarDB cluster.</td>
</tr>
<tr>
<td></td>
<td>• Clone from RDS: clones the data of the selected RDS instance to an ApsaraDB PolarDB cluster.</td>
</tr>
<tr>
<td></td>
<td>• Migration from RDS: clones the data of the selected RDS instance to an ApsaraDB for PolarDB cluster and keeps the data synchronized between the RDS instance and the ApsaraDB for PolarDB cluster. The binlogging feature is enabled for the new cluster by default.</td>
</tr>
<tr>
<td></td>
<td>Select <strong>Clone from RDS</strong>.</td>
</tr>
<tr>
<td>RDS Engine Type</td>
<td>The engine type of the source RDS instance, which cannot be changed.</td>
</tr>
<tr>
<td>RDS Engine Version</td>
<td>The engine version of the source RDS instance, which cannot be changed.</td>
</tr>
<tr>
<td>Source RDS instance</td>
<td>The source RDS instances for selection, which do not include read-only instances.</td>
</tr>
<tr>
<td>Primary availability zone</td>
<td>The zone of the instance. A zone is an independent physical area located within a region. There are no substantive differences between the zones.</td>
</tr>
<tr>
<td></td>
<td>You can deploy the ApsaraDB for PolarDB cluster and the ECS instance in the same zone or in different zones.</td>
</tr>
<tr>
<td>Network Type</td>
<td>The network type of the ApsaraDB for PolarDB cluster, which cannot be changed.</td>
</tr>
<tr>
<td>VPC Vswitch</td>
<td>The VPC and VSwitch to which the ApsaraDB for PolarDB cluster belongs. Make sure that you place your ApsaraDB for PolarDB cluster and the ECS instance to be connected in the same VPC. Otherwise, they cannot communicate with each other through the internal network to achieve optimal performance.</td>
</tr>
<tr>
<td>Database Engine</td>
<td>The database engine of the ApsaraDB for PolarDB cluster, which cannot be changed.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Node Specification</td>
<td>The node specifications of the ApsaraDB for PolarDB cluster. Select the specifications as required. We recommend that you select specifications that are at least the same as those of the source RDS instance. All ApsaraDB for PolarDB nodes are dedicated ones with stable and reliable performance. For more information, see #unique_130.</td>
</tr>
<tr>
<td>Number Nodes</td>
<td>The number of nodes. You do not need to specify this parameter. The system will create a read-only node with the same specifications as those of the primary node by default.</td>
</tr>
<tr>
<td>Storage Cost</td>
<td>The storage capacity. You do not need to specify this parameter. The actual usage is billed hourly in pay-as-you-go mode. For more information, see #unique_130.</td>
</tr>
<tr>
<td>Cluster Name</td>
<td>The cluster name for business distinguishing. The system will automatically create a name for your ApsaraDB for PolarDB cluster if you leave it blank. You can also modify the name after the cluster is created.</td>
</tr>
</tbody>
</table>

5. Specify Duration (only applicable to subscription clusters) and click Buy Now on the right side of the page.

6. Confirm the order information, read the Service Agreement, select the checkbox to agree to it, and click Activate Now.

Next step

Change the database connection point in applications to that of the ApsaraDB for PolarDB cluster as soon as possible. For more information, see #unique_139.

FAQ

Q: Will the source RDS instance be affected when data is cloned from the RDS instance?

A: No, the source RDS instance can run properly.

6.4 Data Migration from a User-created Database to an ApsaraDB RDS MySQL Instance

6.4.1 Migrate data from a user-created MySQL database to an ApsaraDB RDS for MySQL instance

This topic describes how to migrate data from a user-created MySQL database to an ApsaraDB RDS for MySQL instance by using Data Transmission Service (DTS). DTS supports
schema migration, full data migration, and incremental data migration. When you migrate data from a user-created MySQL database, you can select all of the supported migration types to ensure service continuity.

Prerequisites

• An ApsaraDB RDS for MySQL instance is created. For more information, see Create an ApsaraDB RDS for MySQL instance.
• The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
• The available storage space of the destination ApsaraDB RDS for MySQL instance is larger than the total size of the data in the user-created MySQL database.

Precautions

• DTS uses read and write resources of the source and destination databases during full data migration. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before you migrate data, evaluate the performance of the source and destination databases. We recommend that you migrate data during off-peak hours. For example, you can migrate data when the CPU usage of the source and destination databases is less than 30%.
• The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination database.
• DTS uses the ROUND(COLUMN, PRECISION) function to retrieve values from columns of the float or double data type. If the precision is not specified, DTS sets the precision for the float data type to 38 digits and the precision for the double data type to 308 digits. You must check whether the precision settings meet your business requirements.
• DTS automatically creates a destination database in the ApsaraDB RDS for MySQL instance. However, if the name of the source database is invalid, you must manually create a database in the ApsaraDB RDS for MySQL instance before you configure the data migration task.

Note:
For more information about how to create a database and the database naming conventions, see Create databases and accounts.
DTS automatically resumes a failed data migration task. Before you switch your workloads to the destination database, stop or release the data migration task. Otherwise, the data in the source database will overwrite the data in the destination database after the task is resumed.

**Billing**

<table>
<thead>
<tr>
<th>Migration type</th>
<th>Instance configurations</th>
<th>Internet traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema migration and full data migration</td>
<td>Free of charge.</td>
<td>Charged only when data is migrated from Alibaba Cloud over the Internet.</td>
</tr>
<tr>
<td>Incremental data migration</td>
<td>Charged. For more information, see #unique_141.</td>
<td>For more information, see #unique_141.</td>
</tr>
</tbody>
</table>

**Migration types**

- **Schema migration**
  
  DTS migrates the schemas of the required objects to the destination instance. DTS supports schema migration for the following types of objects: table, view, trigger, stored procedure, and function.

  **Note:**
  - During schema migration, DTS changes the value of the SECURITY attribute in views, stored procedures, and functions from DEFINER to INVOKER.
  - DTS does not migrate user information. Before a user can call views, stored procedures, and functions of the destination database, you must grant the read/write permissions to the user.

- **Full data migration**

  DTS migrates historical data of the required objects from the user-created MySQL database to the destination database in the ApsaraDB RDS for MySQL instance.

  **Note:**
  During full data migration, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After full data migration is complete, the tablespace of the destination instance is larger than that of the source database.
- **Incremental data migration**

  After full data migration is complete, DTS retrieves binary log files from the user-created MySQL database. Then, DTS synchronizes incremental data from the user-created MySQL database to the destination ApsaraDB RDS for MySQL instance. Incremental data migration allows you to ensure service continuity when you migrate data from a user-created MySQL database to Alibaba Cloud.

### SQL operations that can be synchronized during incremental data migration

<table>
<thead>
<tr>
<th>Operation type</th>
<th>SQL statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DML</td>
<td>INSERT, UPDATE, DELETE, and REPLACE</td>
</tr>
<tr>
<td>DDL</td>
<td>• ALTER TABLE and ALTER VIEW</td>
</tr>
<tr>
<td></td>
<td>• CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE,</td>
</tr>
<tr>
<td></td>
<td>• CREATE TABLE, and CREATE VIEW</td>
</tr>
<tr>
<td></td>
<td>• DROP INDEX and DROP TABLE</td>
</tr>
<tr>
<td></td>
<td>• RENAME TABLE</td>
</tr>
<tr>
<td></td>
<td>• TRUNCATE TABLE</td>
</tr>
</tbody>
</table>

### Permissions required for database accounts

<table>
<thead>
<tr>
<th>Database</th>
<th>Schema migration</th>
<th>Full data migration</th>
<th>Incremental data migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-created MySQL database</td>
<td>The SELECT</td>
<td>The SELECT</td>
<td>The REPLICATION SLAVE, REPLICA</td>
</tr>
<tr>
<td></td>
<td>permission</td>
<td>permission</td>
<td>TION CLIENT, SHOW VIEW, and SELECT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>permissions</td>
</tr>
<tr>
<td>ApsaraDB RDS for MySQL instance</td>
<td>The read/write permissions</td>
<td>The read/write permissions</td>
<td>The read/write permissions</td>
</tr>
</tbody>
</table>

For information about how to create and authorize a database account, see the following topics:

- #unique_142 for a user-created MySQL database
- Create an account for an RDS for MySQL instance and Change the permissions of an account for an RDS for MySQL instance

**Preparations**

#unique_142
Procedure

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

4. In the upper-right corner of the page, click Create Migration Task.
5. Configure the source and destination databases for the data migration task.

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Task Name</td>
<td>DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Source Database</td>
<td>Instance Type</td>
<td>Select an instance type based on where the source database is deployed. The procedure in this topic uses <strong>User-Created Database with Public IP Address</strong> as an example.</td>
</tr>
<tr>
<td></td>
<td>Note:</td>
<td>If you select other instance types, you must prepare the environments that are required for the source database. For more information, see #unique_143.</td>
</tr>
<tr>
<td>Instance Region</td>
<td>If the instance type is set to <strong>User-Created Database with Public IP Address</strong>, you do not need to specify the <strong>instance region</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note:</td>
<td>If a whitelist is configured for the user-created MySQL database, you must manually add the CIDR blocks of DTS servers to the whitelist of the user-created MySQL database. You can click <strong>Get IP Address Segment of DTS</strong> next to <strong>Instance Region</strong> to obtain the CIDR blocks of DTS servers.</td>
</tr>
<tr>
<td>Database Type</td>
<td>Select MySQL</td>
<td></td>
</tr>
<tr>
<td>Hostname or IP Address</td>
<td>Enter the IP address that is used to access the user-created MySQL database. In this example, enter the public IP address.</td>
<td></td>
</tr>
<tr>
<td>Port Number</td>
<td>Enter the service port number of the user-created MySQL database. The default port number is <strong>3306</strong>.</td>
<td></td>
</tr>
<tr>
<td>Database Account</td>
<td>Enter the account of the user-created MySQL database. For more information about the permissions that are required for the account, see <strong>Permissions required for database accounts</strong>.</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the source database account. Note: After you specify the source database parameters, click Test Connectivity next to Database Password to verify whether the specified parameters are valid. If the specified parameters are valid, the Passed message appears. If the Failed message appears, click Check next to Failed. Modify the source database parameters based on the check results.</td>
</tr>
<tr>
<td>Destination Database</td>
<td>Instance Type</td>
<td>Select RDS Instance.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>Select the region where the destination RDS instance resides.</td>
</tr>
<tr>
<td></td>
<td>RDS Instance ID</td>
<td>Select the ID of the destination RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account of the destination RDS instance. For more information about the permissions that are required for the account, see Permissions required for database accounts.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the destination database account. Note: After you specify the destination database parameters, click Test Connectivity next to Database Password to verify whether the parameters are valid. If the specified parameters are valid, the Passed message appears. If the Failed message appears, click Check next to Failed. Modify the destination database parameters based on the check results.</td>
</tr>
<tr>
<td></td>
<td>Encryption</td>
<td>Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before you configure the data migration task. For more information, see Configure SSL encryption for an RDS for MySQL instance. Note: The Encryption parameter is available only for regions in mainland China and the Hong Kong (China) region.</td>
</tr>
</tbody>
</table>
6. In the lower-right corner of the page, click **Set Whitelist and Next**.

**Note:**
The CIDR blocks of DTS servers are automatically added to the whitelist of the destination RDS instance. This ensures that DTS servers can connect to the destination RDS instance.

7. Select the migration types and objects to be migrated.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Migration Types | • To perform only full data migration, select **Schema Migration** and **Full Data Migration**.  
                  • To migrate data with minimal downtime, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. |

**Note:**
If **Incremental Data Migration** is not selected, do not write data into the source database during full data migration. This ensures data consistency between the source and destination databases.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
<td>Select objects from the Available section and click the icon to move the objects to the Selected section.</td>
</tr>
</tbody>
</table>

**Note:**
- You can select columns, tables, or databases as the objects to be migrated.
- After an object is migrated to the destination database, the name of the object remains the same as that in the source database. You can change the names of the objects that are migrated to the destination database by using the object name mapping feature. For more information about how to use this feature, see #unique_144.
- If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.

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

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

9. After the task passes the precheck, click Next.

10. In the Confirm Settings dialog box, specify the Channel Specification and select the Data Transmission Service (Pay-As-You-Go) Service Terms.

11. Click Buy and Start to start the migration task.

**Stop the migration task**

**Warning:**
We recommend that you prepare a rollback solution to migrate incremental data from the destination database to the source database in real time. This allows you to minimize the negative impact of switching your workloads to the destination database. For more information, see #unique_145. If you do not need to switch your workloads, you can stop the migration task by using the following procedure.
• Full data migration

Do not manually stop a task during full data migration. Otherwise, the system may fail to migrate all data. Wait until the migration task automatically ends.

• Incremental data migration

The task does not automatically end during incremental data migration. You must manually stop the migration task.

1. Wait until the task progress bar shows **Incremental Data Migration** and **The migration task is not delayed**. Then, stop writing data to the source database for a few minutes. In some cases, the progress bar shows the delay time of **incremental data migration**.

2. After the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.

What to do next

The database accounts used for data migration have the read/write permissions. After data is migrated, you must delete the database accounts to ensure security.

### 6.4.2 Migrate data from a user-created Oracle database to an ApsaraDB RDS for MySQL instance

This topic describes how to migrate data from a user-created Oracle database to an ApsaraDB RDS for MySQL instance by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. When you migrate data from a user-created Oracle database, you can select all of the supported migration types to ensure service continuity.

**Prerequisites**

- The version of the user-created Oracle database is 9i, 10g, 11g, 12c, 18c, or 19c.
- Supplemental logging, including SUPPLEMENTAL_LOG_DATA_PK and SUPPLEMENTAL_LOG_DATA_UI, is enabled for the user-created Oracle database. For more information, see [Supplemental Logging](#).
• The ARCHIVELOG mode is enabled for the user-created Oracle database. Archived log files are accessible and a suitable retention period is set for archived log files. For more information, see Managing Archived Redo Log Files.

• The service port of the user-created Oracle database is accessible over the Internet.

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

Precautions

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

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

• Table names in the ApsaraDB RDS for MySQL instance are case-insensitive. If a table name in the user-created Oracle database contains uppercase letters, ApsaraDB RDS for MySQL converts all uppercase letters to lowercase letters before creating the table. If the source Oracle database contains identical table names that differ only in capitalization, these table names are identified as duplicate. During schema migration, the following message is returned: "The object already exists". To avoid name conflicts in the destination database, you can change the names of the migrated objects by using the object name mapping feature. For more information, see #unique_144.

• DTS automatically creates a destination database in the ApsaraDB RDS for MySQL instance. However, if the name of the source database is invalid, you must manually create a database in the ApsaraDB RDS for MySQL instance before you configure the data migration task. For more information about how to create a database and the database naming conventions, see Create a database.
Billing

<table>
<thead>
<tr>
<th>Migration type</th>
<th>Instance configurations</th>
<th>Internet traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema migration and full data migration</td>
<td>Free of charge.</td>
<td>Charged only when data is migrated from Alibaba Cloud over the Internet. For more information, see #unique_141.</td>
</tr>
<tr>
<td>Incremental data migration</td>
<td>Charged. For more information, see #unique_141.</td>
<td></td>
</tr>
</tbody>
</table>

Migration types

- Schema migration

  DTS supports schema migration for tables and indexes. DTS does not support schema migration for the following types of objects: view, synonym, trigger, stored procedure, function, package, and user-defined type. DTS has the following limits on schema migration for tables and indexes:
  
  - Schema migration of nested tables is not supported. Clustered tables and index-organized tables (IOTs) are converted into common tables in the destination database.
  
  - Schema migration of function-based indexes, domain indexes, bitmap indexes, and reverse indexes is not supported.

- Full data migration

  DTS migrates historical data of the required objects from the user-created Oracle database to the destination ApsaraDB RDS for MySQL instance.

- Incremental data migration

  DTS retrieves redo log files from the user-created Oracle database. Then, DTS synchronizes incremental data from the user-created Oracle database to the destination database in the ApsaraDB RDS for MySQL instance. Incremental data migration allows you to ensure service continuity when you migrate data from the user-created Oracle database to the destination database.

**SQL operations that can be synchronized during incremental data migration**

- INSERT, DELETE, and UPDATE
• CREATE TABLE

**Note:**
The CREATE TABLE operations to create tables that contain functions cannot be synchronized.

• ALTER TABLE, ADD COLUMN, DROP COLUMN, RENAME COLUMN, and ADD INDEX
• DROP TABLE
• RENAME TABLE, TRUNCATE TABLE, and CREATE INDEX

**Data type mappings**

For more information, see #unique_146.

**Preparations**

Log on to the source Oracle database, create an account for data collection, and grant permissions to the account.

**Note:**
If you have created a database account and the account has the permissions that are listed in the following table, skip this step.

<table>
<thead>
<tr>
<th>Database</th>
<th>Schema migration</th>
<th>Full data migration</th>
<th>Incremental data migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-created Oracle database</td>
<td>The owner permission on schemas</td>
<td>The owner permission on schemas</td>
<td>The database administrator (DBA) permission</td>
</tr>
<tr>
<td>ApsaraDB RDS for MySQL instance</td>
<td>The write permission on the destination database</td>
<td>The write permission on the destination database</td>
<td>The write permission on the destination database</td>
</tr>
</tbody>
</table>

For more information about how to create and authorize a database account, see the following topics:

• **CREATE USER** and **GRANT** for a user-created Oracle database
• **Create an account for an RDS for MySQL instance** and **Change the permissions of an account for an RDS for MySQL instance**
If you want to migrate incremental data from an Oracle database but the DBA permission cannot be granted to the database account, you can grant fine-grained permissions to the account. The following sample statements show you how to grant specific permissions to an Oracle database account.

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click Data Migration.
3. At the top of the Migration Tasks page, select the region where the destination RDS instance resides.
4. In the upper-right corner of the page, click Create Migration Task.
5. Configure the source and destination databases for the data migration task.

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Task Name</td>
<td>DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Instance Type</td>
<td>Select an instance type based on where the source database is deployed. The procedure in this topic uses <strong>User-Created Database with Public IP Address</strong> as an example.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> If you select other instance types, you must prepare the environments that are required for the source database. For more information, see #unique_143.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Instance Region</td>
<td></td>
<td>If the instance type is set to <strong>User-Created Database with Public IP Address</strong>, you do not need to specify the <strong>instance region</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> If a whitelist is configured for the user-created Oracle database, you must manually add the CIDR blocks of DTS servers to the whitelist of the user-created Oracle database. You can click <strong>Get IP Address Segment of DTS</strong> next to <strong>Instance Region</strong> to obtain the CIDR blocks of DTS servers.</td>
</tr>
<tr>
<td>Database Type</td>
<td>Select <strong>Oracle</strong>.</td>
<td></td>
</tr>
<tr>
<td>Hostname or IP Address</td>
<td>Enter the IP address that is used to connect to the user-created Oracle database. In this example, enter the public IP address.</td>
<td></td>
</tr>
<tr>
<td>Port Number</td>
<td>Enter the service port number of the user-created Oracle database. The default port number is <strong>1521</strong>.</td>
<td></td>
</tr>
<tr>
<td>Instance Type</td>
<td><strong>Non-RAC Instance</strong>: If you select this option, you must specify the <strong>SID</strong>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>RAC Instance</strong>: If you select this option, you must specify the <strong>Service Name</strong>.</td>
<td></td>
</tr>
<tr>
<td>Database Account</td>
<td>Enter the account of the user-created Oracle database. For more information about the permissions that are required for the account, see <strong>Preparations</strong>.</td>
<td></td>
</tr>
<tr>
<td>Database Password</td>
<td>Enter the password for the source database account.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> After you specify the source database parameters, click <strong>Test Connectivity</strong> next to <strong>Database Password</strong> to verify whether the specified parameters are valid. If the specified parameters are valid, the <strong>Passed</strong> message appears. If the <strong>Failed</strong> message appears, click <strong>Check</strong> next to <strong>Failed</strong>. Modify the source database parameters based on the check results.</td>
<td></td>
</tr>
<tr>
<td>Destination Database</td>
<td>Select <strong>RDS Instance</strong>.</td>
<td></td>
</tr>
<tr>
<td>Instance Region</td>
<td>Select the region where the destination RDS instance resides.</td>
<td></td>
</tr>
</tbody>
</table>
## Parameter Description

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDS Instance ID</td>
<td></td>
<td>Select the ID of the destination RDS instance.</td>
</tr>
<tr>
<td>Database Account</td>
<td></td>
<td>Enter the database account of the destination RDS instance. For more information about the permissions that are required for the account, see Preparations.</td>
</tr>
<tr>
<td>Database Password</td>
<td></td>
<td>Enter the password for the destination database account.</td>
</tr>
</tbody>
</table>

**Note:**
After you specify the destination database parameters, click **Test Connectivity** next to **Database Password** to verify whether the parameters are valid. If the specified parameters are valid, the **Passed** message appears. If the **Failed** message appears, click **Check** next to **Failed**. Modify the destination database parameters based on the check results.

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

**Note:**
The CIDR blocks of DTS servers are automatically added to the whitelist of the destination RDS instance. This ensures that DTS servers can connect to the destination RDS instance.
7. Select the migration types and objects to be migrated.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Migration Types  | • To perform only full data migration, select **Schema Migration** and **Full Data Migration**.  
• To migrate data with minimal downtime, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. |

**Note:**  
If **Incremental Data Migration** is not selected, do not write data into the source database during full data migration. This ensures data consistency between the source and destination databases.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
<td>Select objects from the Available section and click the icon to move the objects to the Selected section.</td>
</tr>
</tbody>
</table>

**Note:**
- You can select columns, tables, or databases as the objects to be migrated.
- After an object is migrated to the destination RDS instance, the name of the object remains the same as that in the user-created Oracle database. You can change the names of the objects that are migrated to the destination RDS instance by using the object name mapping feature. For more information about how to use this feature, see #unique_144.

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

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

9. After the task passes the precheck, click **Next**.

10. In the **Confirm Settings** dialog box, specify the **Channel Specification** and select the **Data Transmission Service (Pay-As-You-Go) Service Terms**.

11. Click **Buy and Start** to start the migration task.

- **Full data migration**
  
  Do not manually stop a task during full data migration. Otherwise, data migrated to the destination database will be incomplete. Wait until the migration task automatically stops.

- **Incremental data migration**
  
  An incremental data migration task does not automatically stop. You must manually stop the migration task.

**Note:**
Select an appropriate time to manually stop the migration task. For example, you can stop the migration task during off-peak hours or before you switch your workloads to the destination instance.

a. Wait until **Incremental Data Migration** and **The migration task is not delayed** appear in the progress bar of the migration task. Then, stop writing data to the source database for a few minutes. The delay time of **incremental data migration** may be displayed in the progress bar.

b. After the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.

12. Switch your workloads to the destination RDS instance.

**What to do next**

The database accounts used for data migration have the read/write permissions. After the data migration is complete, you must delete the accounts of both the user-created Oracle database and the ApsaraDB RDS for MySQL instance to ensure database security.

**More information**

DTS supports reverse data transmission when you migrate data from a user-created Oracle database to an ApsaraDB RDS for MySQL instance. You can use this feature to synchronize data changes from the ApsaraDB RDS for MySQL instance to the user-created Oracle database. To do this, submit a ticket.

### 6.4.3 Migrate data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an ApsaraDB RDS for MySQL database

This topic describes how to migrate data from a user-created MySQL database that is connected over Express Connect, VPN Gateway, or Smart Access Gateway to an ApsaraDB RDS for MySQL database by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. To migrate data from a user-created MySQL database, you can select all of the supported migration types to ensure service continuity.
Prerequisites

- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The available storage space of the destination ApsaraDB RDS for MySQL database is larger than the total space of the data in the user-created MySQL database.
- The on-premises network to which the user-created MySQL database belongs is connected to Alibaba Cloud over Express Connect, VPN Gateway, or Smart Access Gateway.

Note:
For more information, see #unique_147.

Precautions

- DTS uses read and write resources of the source and destination databases during full data migration. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before you migrate data, evaluate the performance of the source and destination databases. We recommend that you migrate data during off-peak hours. For example, you can migrate data when the CPU usage of the source and destination databases is less than 30%.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination database.
- DTS uses the ROUND(COLUMN,PRECISION) function to retrieve values from columns of the float or double data type. If the precision is not specified, DTS sets the precision for the float data type to 38 digits and the precision for the double data type to 308 digits. You must check whether the precision settings meet your business requirements.
- DTS automatically creates a destination database in the ApsaraDB RDS for MySQL instance. However, if the name of the source database is invalid, you must manually create a database in the ApsaraDB RDS for MySQL instance before you configure the data migration task.

Note:
For more information about how to create a database and the database naming conventions, see Create databases and accounts.

- DTS automatically resumes a failed data migration task. Before you switch your workloads to the destination database, stop or release the data migration task. Otherwise, the data in the source database will overwrite the data in the destination database after the task is resumed.

**Billing**

<table>
<thead>
<tr>
<th>Migration type</th>
<th>Instance configurations</th>
<th>Internet traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema migration and full data migration</td>
<td>Free of charge.</td>
<td>Charged only when data is migrated from Alibaba Cloud over the Internet. For more information, see #unique_141.</td>
</tr>
<tr>
<td>Incremental data migration</td>
<td>Charged. For more information, see #unique_141.</td>
<td>For more information, see #unique_141.</td>
</tr>
</tbody>
</table>

**Migration types**

- **Schema migration**

  DTS migrates the schemas of the required objects to the destination instance. DTS supports schema migration for the following types of objects: table, view, trigger, stored procedure, and function.

  **Note:**
  - During schema migration, DTS changes the value of the SECURITY attribute in views, stored procedures, and functions from DEFINER to INVOKER.
  - DTS does not migrate user information. Before a user can call views, stored procedures, and functions of the destination database, you must grant the read/write permissions to the user.

- **Full data migration**

  DTS migrates historical data of the required objects from the user-created MySQL database to the destination database in the ApsaraDB RDS for MySQL instance.

  **Note:**
During full data migration, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After full data migration is complete, the tablespace of the destination instance is larger than that of the source database.

• Incremental data migration

After full data migration is complete, DTS retrieves binary log files from the user-created MySQL database. Then, DTS synchronizes incremental data from the user-created MySQL database to the destination ApsaraDB RDS for MySQL instance. Incremental data migration allows you to ensure service continuity when you migrate data from a user-created MySQL database to Alibaba Cloud.

**SQL operations that can be synchronized during incremental data migration**

<table>
<thead>
<tr>
<th>Operation type</th>
<th>SQL statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DML</td>
<td>INSERT, UPDATE, DELETE, and REPLACE</td>
</tr>
</tbody>
</table>
| DDL            | • ALTER TABLE and ALTER VIEW  
                           • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW  
                           • DROP INDEX and DROP TABLE  
                           • RENAME TABLE  
                           • TRUNCATE TABLE |

**Permissions required for database accounts**

<table>
<thead>
<tr>
<th>Database</th>
<th>Schema migration</th>
<th>Full data migration</th>
<th>Incremental data migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-created MySQL database</td>
<td>The SELECT permission</td>
<td>The SELECT permission</td>
<td>The REPLICATION SLAVE, REPLICATION CLIENT, SHOW VIEW, and SELECT permissions</td>
</tr>
<tr>
<td>ApsaraDB RDS for MySQL instance</td>
<td>The read/write permissions</td>
<td>The read/write permissions</td>
<td>The read/write permissions</td>
</tr>
</tbody>
</table>

For more information about how to create and authorize a database account, see the following topics:

• #unique_142 for a user-created MySQL database
- Create an account for an RDS for MySQL instance and Change the permissions of an account for an RDS for MySQL instance

**Preparations**

1. #unique_142.
2. #unique_148.

**Procedure**

1. Log on to the DTS console.
2. In the left-side navigation pane, click **Data Migration**.
3. At the top of **Migration Tasks** the page, select the region where the destination cluster resides.
4. In the upper-right corner of the page, click **Create Migration Task**.
5. Configure the information about the source and destination databases for the data migration task.

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Task Name</td>
<td>DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Instance Type</td>
<td>Select <strong>User-Created Database Connected Over Express Connect, VPN Gateway, or Smart Access Gateway</strong>.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>Select the region to which the VPC that is connected to Express Connect, VPN Gateway, or Smart Access Gateway belongs.</td>
</tr>
<tr>
<td></td>
<td>Peer VPC</td>
<td>Select the VPC that is connected to Express Connect, VPN Gateway, or Smart Access Gateway.</td>
</tr>
<tr>
<td></td>
<td>Database Type</td>
<td>Select <strong>MySQL</strong>.</td>
</tr>
<tr>
<td></td>
<td>IP Address</td>
<td>Enter the endpoint that is used to connect to the user-created MySQL database.</td>
</tr>
<tr>
<td></td>
<td>Port Number</td>
<td>Enter the service port number of the user-created MySQL database. The default port number is <strong>3306</strong>.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Database Account</td>
<td>Database Account</td>
<td>Enter the account for the user-created MySQL database. For more information about permissions required for the account, see <a href="#">Permissions required for database accounts</a>.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> After the source database parameters are specified, click <a href="#">Test Connectivity</a> next to the <strong>Database Password</strong> parameter to verify whether the specified parameters are correct. If the source database parameters are correct, the <strong>Test Passed</strong> message is displayed, if the <strong>Test Failed</strong> message is displayed, click <a href="#">Diagnose</a> in the <strong>Test Failed</strong> message. Modify the source database parameters as prompted.</td>
</tr>
<tr>
<td>Destination Database</td>
<td>Instance Type</td>
<td>Select <strong>RDS Instance</strong>.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>Select the region where the destination RDS instance resides.</td>
</tr>
<tr>
<td></td>
<td>RDS Instance ID</td>
<td>Select the ID of the destination RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account of the destination RDS instance. For more information about permissions required for the account, see <a href="#">Permissions required for database accounts</a>.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> After the destination database parameters are specified, click <a href="#">Test Connectivity</a> next to the <strong>Database Password</strong> parameter to verify whether the specified parameters are correct. If the destination database parameters are correct, the <strong>Test Passed</strong> message is displayed, if the <strong>Test Failed</strong> message is displayed, click <a href="#">Diagnose</a> in the <strong>Test Failed</strong> message. Modify the destination database parameters as prompted.</td>
</tr>
</tbody>
</table>
ApsaraDB for RDS

RDS MySQL Database / 6 Data Migration

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption</td>
<td>Select <strong>Non-encrypted</strong> or <strong>SSL-encrypted</strong>. If you want to select <strong>SSL-encrypted</strong>, you must enable SSL encryption for the RDS instance before configuring the data migration task. For more information, see Configure SSL encryption for an RDS for MySQL instance.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
The *Encryption* parameter is available only in mainland China and Hong Kong(China). |

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

**Note:**
The CIDR blocks of DTS servers are automatically added to the whitelist of the destination ApsaraDB RDS for MySQL instance. This ensures that DTS servers can connect to the destination RDS instance.
7. Configure migration types and objects.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
</table>
| Migration types | • To perform only full data migration, select **Schema Migration** and **Full Data Migration**.  
|               | • If you want to migrate data without business disruptions, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**.  
<p>|               | <strong>Note:</strong> If <strong>Incremental Data Migration</strong> is not selected, do not write data into the source database during full data migration. This ensures data consistency between the source and destination databases. |</p>
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects to be migrated</td>
<td>Select the objects to be migrated in the <strong>Available</strong> section and click icon to move them to the <strong>Selected</strong> section.</td>
</tr>
</tbody>
</table>

**Note:**
- Objects to be migrated can be databases, tables, or columns.
- By default, the selected objects are not renamed after the migration. If you want to rename the objects that are migrated to the destination instance, you can use the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_144.
- If you use the object name mapping feature for an object, objects that depend on the object may fail to be migrated.

8. Click **Precheck** on the lower right of the page.

**Note:**
- A precheck is performed for a data migration task. A data migration task can be started only if it passes the precheck.
- If the precheck fails, click icon corresponding to each failed item to view the details. Fix the problems as instructed and run the precheck again.

9. After the precheck is passed, click **Next**.

10. On the **Confirm Settings** dialog box that appears, specify **Channel Specification** and select the **Data Transmission Service (Pay-As-You-Go) Service Terms**.
11. Click **Buy and Start** to start the data migration task.

- Schema migration and full data migration

  Do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the data migration task stops when it is complete.

- Schema migration, full data migration, and incremental data migration

  An incremental data migration task does not automatically end. You must manually end the migration task.

  **Note:**
  Select an appropriate time point to manually end the migration task. For example, you can end the migration task during off-peak hours or before you switch your workloads to the destination cluster.

  a. When the task progress bar switches to **Incremental Data Migration** and the message **The migration task is not delayed** appears, stop writing new data to the source database for a few minutes. Then, the progress bar will show the latency of the incremental data migration.

  b. When the status of incremental data migration changes to **The migration task is not delayed**, manually stop the migration task.

12. Switch your workloads to the ApsaraDB RDS for MySQL instance.

### 6.4.4 Migrate data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an ApsaraDB RDS for MySQL instance across Alibaba Cloud accounts

This topic describes how to migrate data from a user-created MySQL database that is connected over Express Connect, VPN Gateway, or Smart Access Gateway to an ApsaraDB RDS for MySQL instance by using Data Transmission Service (DTS). In this scenario, the Express Connect circuit and the destination RDS instance are owned by different Alibaba
Cloud accounts. DTS supports schema migration, full data migration, and incremental data migration. You can select all of the supported migration types to ensure service continuity.

**Prerequisites**

- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The available storage space of the destination ApsaraDB RDS for MySQL instance is larger than the total size of the data in the user-created MySQL database.
- The on-premises network to which the user-created MySQL database belongs is connected to Alibaba Cloud VPC over Express Connect, VPN Gateway, or Smart Access Gateway. The Express Connect circuit and the destination RDS instance are owned by different Alibaba Cloud accounts.

**Note:**

For more information about how to connect an on-premises data center to a VPC, see #unique_147.

**Context**

The on-premises data center is connected to Alibaba Cloud VPC over Express Connect, VPN Gateway, or Smart Access Gateway. You need to migrate data from a user-created MySQL database that resides in an on-premises data center to an RDS instance across different Alibaba Cloud accounts. The following figure shows the architecture for this scenario.
Precautions

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

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

- DTS uses the `ROUND(COLUMN,PRECISION)` function to retrieve values from columns of the float or double data type. If the precision is not specified, DTS sets the precision for the float data type to 38 digits and the precision for the double data type to 308 digits. You must check whether the precision settings meet your business requirements.

- DTS automatically creates a destination database in the ApsaraDB RDS for MySQL instance. However, if the name of the source database is invalid, you must manually create a database in the ApsaraDB RDS for MySQL instance before you configure the data migration task.

Note:
For more information about how to create a database and the database naming conventions, see Create databases and accounts.

- DTS automatically resumes a failed data migration task. Before you switch your workloads to the destination database, stop or release the data migration task. Otherwise, the data in the source database will overwrite the data in the destination database after the task is resumed.
Billing

<table>
<thead>
<tr>
<th>Migration type</th>
<th>Instance configurations</th>
<th>Internet traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema migration and full data migration</td>
<td>Free of charge.</td>
<td>Charged only when data is migrated from Alibaba Cloud over the Internet. For more information, see #unique_141.</td>
</tr>
<tr>
<td>Incremental data migration</td>
<td>Charged. For more information, see #unique_141.</td>
<td></td>
</tr>
</tbody>
</table>

Migration types

- **Schema migration**

  DTS migrates the schemas of the required objects to the destination instance. DTS supports schema migration for the following types of objects: table, view, trigger, stored procedure, and function.

  **Note:**
  - During schema migration, DTS changes the value of the SECURITY attribute in views, stored procedures, and functions from DEFINER to INVOKER.
  - DTS does not migrate user information. Before a user can call views, stored procedures, and functions of the destination database, you must grant the read/write permissions to the user.

- **Full data migration**

  DTS migrates historical data of the required objects from the user-created MySQL database to the destination database in the ApsaraDB RDS for MySQL instance.

  **Note:**
  During full data migration, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After full data migration is complete, the tablespace of the destination instance is larger than that of the source database.

- **Incremental data migration**

  After full data migration is complete, DTS retrieves binary log files from the user-created MySQL database. Then, DTS synchronizes incremental data from the user-created MySQL database to the destination ApsaraDB RDS for MySQL instance. Incremental data
migrations allows you to ensure service continuity when you migrate data from a user-
created MySQL database to Alibaba Cloud.

SQL operations that can be synchronized during incremental data migration

<table>
<thead>
<tr>
<th>Operation type</th>
<th>SQL statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DML</td>
<td>INSERT, UPDATE, DELETE, and REPLACE</td>
</tr>
<tr>
<td>DDL</td>
<td>ALTER TABLE and ALTER VIEW, CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW, DROP INDEX and DROP TABLE, RENAME TABLE, TRUNCATE TABLE</td>
</tr>
</tbody>
</table>

Permissions required for database accounts

<table>
<thead>
<tr>
<th>Database</th>
<th>Schema migration</th>
<th>Full data migration</th>
<th>Incremental data migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-created MySQL database</td>
<td>The SELECT permission</td>
<td>The SELECT permission</td>
<td>The REPLICATION SLAVE, REPLICATION CLIENT, SHOW VIEW, and SELECT permissions</td>
</tr>
<tr>
<td>ApsaraDB RDS for MySQL instance</td>
<td>The read/write permissions</td>
<td>The read/write permissions</td>
<td>The read/write permissions</td>
</tr>
</tbody>
</table>

For more information about how to create and authorize a database account, see the following topics:

- #unique_142 for a user-created MySQL database
- Create an account for an RDS for MySQL instance and Change the permissions of an account for an RDS for MySQL instance

Preparations

1. #unique_142.
2. Log on to the Alibaba Cloud console by using the Alibaba Cloud account to which the Express Connect circuit belongs. Authorize DTS to access the network that is connected over Express Connect. For more information, see #unique_150.
3. Create a RAM role and authorize this role to access the cloud resources of the Alibaba Cloud account to which the Express Connect circuit belongs. For more information, see #unique_151.

Procedure

1. Log on to the DTS console by using the Alibaba Cloud account to which the destination RDS instance belongs.
2. In the left-side navigation pane, click Data Migration.
3. At the top of Migration Tasks the page, select the region where the destination cluster resides.
4. In the upper-right corner of the page, click Create Migration Task.
5. Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway as the instance type. Then, click VPC of Another Alibaba Cloud Account next to the Peer VPC field.
6. Configure the source and destination databases for the data migration task.

### Table: Data Migration Configuration Parameters

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Task Name</td>
<td>DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Instance Type</td>
<td>Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>Select the region of the VPC that is connected to the user-created MySQL database.</td>
</tr>
<tr>
<td></td>
<td>RDS Instance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RDS Instance ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Encryption</td>
<td></td>
</tr>
<tr>
<td>Destination</td>
<td>Instance Type</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Encryption</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Alibaba Cloud</td>
<td>Account ID</td>
<td>Enter the ID of the Alibaba Cloud account to which the Express Connect circuit belongs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> To obtain the ID of the Alibaba Cloud account to which the Express Connect circuit belongs, you must log on to the Account Management console by using this account. The account ID is displayed on the Security Settings page.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><img src="image" alt="Account Management" /></td>
</tr>
<tr>
<td>Role Name</td>
<td></td>
<td>Enter the name of the RAM role that you created earlier in Preparations.</td>
</tr>
<tr>
<td>Peer VPC</td>
<td></td>
<td>Select the ID of the VPC that is connected to the user-created MySQL database.</td>
</tr>
<tr>
<td>Database Type</td>
<td></td>
<td>Select MySQL.</td>
</tr>
<tr>
<td>IP Address</td>
<td></td>
<td>Enter the endpoint that is used to access the user-created MySQL database.</td>
</tr>
<tr>
<td>Port Number</td>
<td></td>
<td>Enter the service port number of the user-created MySQL database. The default port number is 3306.</td>
</tr>
<tr>
<td>Database Account</td>
<td></td>
<td>Enter the account of the user-created MySQL database. For more information about the permissions that are required for the account, see Permissions required for database accounts.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password of the source database account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> After you specify the source database parameters, click <strong>Test Connectivity</strong> next to <strong>Database Password</strong> to verify whether the specified parameters are valid. If the specified parameters are valid, the <strong>Passed</strong> message appears. If the <strong>Failed</strong> message appears, click <strong>Check</strong> next to <strong>Failed</strong>. Modify the source database parameters based on the check results.</td>
</tr>
<tr>
<td>Destination Database</td>
<td>Instance Type</td>
<td>Select <strong>RDS Instance</strong>.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>Select the region where the destination RDS instance resides.</td>
</tr>
<tr>
<td></td>
<td>RDS Instance ID</td>
<td>Select the ID of the destination RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account of the destination RDS instance. For more information about the permissions that are required for the account, see Permissions required for database accounts.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password of the destination database account.</td>
</tr>
<tr>
<td></td>
<td>Encryption</td>
<td>Select <strong>Non-encrypted</strong> or <strong>SSL-encrypted</strong>. If you want to select <strong>SSL-encrypted</strong>, you must enable SSL encryption for the RDS instance before you configure the data migration task. For more information, see Configure SSL encryption for an RDS MySQL instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The <strong>Encryption</strong> parameter is available only for regions in mainland China and the Hong Kong (China) region.</td>
</tr>
</tbody>
</table>
7. In the lower-right corner of the page, click **Set Whitelist and Next**.

**Note:**
DTS adds the CIDR blocks of DTS servers to the whitelist of the destination ApsaraDB RDS for MySQL instance. This ensures that DTS servers can connect to the destination RDS instance.

8. Configure migration types and objects.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migration types</td>
<td>• To perform only full data migration, select <strong>Schema Migration</strong> and <strong>Full Data Migration</strong>.</td>
</tr>
<tr>
<td></td>
<td>• If you want to migrate data without disruptions to your business, select <strong>Schema Migration</strong>, <strong>Full Data Migration</strong>, and <strong>Incremental Data Migration</strong>.</td>
</tr>
</tbody>
</table>

**Note:**
If **Incremental Data Migration** is not selected, do not write data into the source database during full data migration. This ensures data consistency between the source and destination databases.
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects to be migrated</td>
<td>Select the objects to be migrated in the Available section and click icon to move them to the Selected section.</td>
</tr>
</tbody>
</table>
| Note:                   | • Objects to be migrated can be databases, tables, or columns.  
                          • The selected objects are not renamed after the migration by default. If you want to rename the objects migrated to the destination instance, you can use the object name mapping feature provided by DTS. For more information, see #unique_144.  
                          • If you use the object name mapping feature for an object, objects that depend on the object may fail to be migrated. |

**9.** Click Precheck on the lower right of the page.

**Note:**
- A precheck is performed for a data migration task. A data migration task can be started only if it passes the precheck.  
- If the precheck fails, click icon corresponding to each failed item to view the details. Fix the problems as instructed and run the precheck again.

**10.** After the precheck is passed, click Next.

**11.** On the Confirm Settings dialog box that appears, specify Channel Specification and select the Data Transmission Service (Pay-As-You-Go) Service Terms.

**12.** Click Buy and Start to start the data migration task.

- Schema migration and full data migration
  - Do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the data migration task stops when it is complete.
- Schema migration, full data migration, and incremental data migration
  - An incremental data migration task does not automatically end. You must manually end the migration task.

**Note:**
Select an appropriate time point to manually end the migration task. For example, you can end the migration task during off-peak hours or before you switch your workloads to the destination cluster.

a. When the task progress bar switches to **Incremental Data Migration** and the message **The migration task is not delayed** appears, stop writing new data to the source database for a few minutes. Then, the progress bar will show the latency of the **incremental data migration**.

b. When the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.

13. Switch your workloads to the destination ApsaraDB RDS for MySQL instance.

### 6.4.5 Migrate data from a user-created Db2 database to an ApsaraDB RDS for MySQL database

This topic describes how to migrate data from a user-created Db2 database to an ApsaraDB RDS for MySQL database by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. To migrate data from a user-created Db2 database, you can select all of the supported migration types to ensure service continuity.

#### Prerequisites

- The Db2 database is deployed on Linux.
- The version of the Db2 database is 9.7 or 10.5.
- The available storage space of the destination ApsaraDB RDS for MySQL database is larger than the total space of the data in the user-created Db2 database.

#### Notes

- DDL operations cannot be synchronized.
- If the name of the source database is invalid, you must create a database in the ApsaraDB RDS for MySQL instance before configuring a data migration task.
For more information about how to create a database and the database naming conventions, see Create databases and accounts.

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

- DTS automatically resumes a failed data migration task. Before switching your workloads to the destination instance, you must stop or release the data migration task. Otherwise, the data from the source database will overwrite the data in the destination instance after the task is resumed.

### Billing

<table>
<thead>
<tr>
<th>Migration type</th>
<th>Migration channel fee</th>
<th>Public network traffic fee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema migration or full data migration</td>
<td>Free of charge</td>
<td>Migrating data from Alibaba Cloud over the Internet incurs fees. For more information, see #unique_141.</td>
</tr>
<tr>
<td>Incremental data migration</td>
<td>Billed. For more information, see #unique_141.</td>
<td></td>
</tr>
</tbody>
</table>

### Migration types

- **Schema migration**

  DTS migrates the schemas of the required objects to the destination instance. DTS supports schema migration for the following types of objects: table, index, and foreign key.

- **Full data migration**

  DTS migrates historical data of the required objects from the source Db2 database to the destination ApsaraDB RDS for MySQL database.
• Incremental data migration

After full data migration is complete, DTS synchronizes incremental data from the user-created Db2 database to the destination ApsaraDB RDS for MySQL database. Incremental data migration helps you ensure service continuity when you migrate data from a user-created Db2 database.

Permissions required for database accounts

<table>
<thead>
<tr>
<th>Database</th>
<th>Schema migration</th>
<th>Full data migration</th>
<th>Incremental data migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-created Db2 database</td>
<td>The CONNECT and SELECT permissions</td>
<td>The CONNECT and SELECT permissions</td>
<td>The DBADM permission</td>
</tr>
<tr>
<td>ApsaraDB RDS for MySQL database</td>
<td>The read/write permissions</td>
<td>The read/write permissions</td>
<td>The read/write permissions</td>
</tr>
</tbody>
</table>

For more information about how to create and authorize a database account, see the following topics:

• Creating group and user IDs for a Db2 database installation (Linux and UNIX) and Authorities overview for a user-created Db2 database
• Create an account for an RDS for MySQL instance and Change the permissions of an account for an RDS for MySQL instance for an ApsaraDB RDS for MySQL database

Data migration process

To avoid data migration failures caused by dependencies between objects, DTS migrates the schemas and data of the source Db2 database in the following order:

1. Migrate the schemas and indexes.
2. Perform full data migration.
3. Migrate the schemas of foreign keys.
4. Perform incremental data migration.

Preparations before incremental data migration

Before configuring an incremental data migration task, you must enable the archive log feature of the source Db2 database. For more information, see Primary log archive method and Secondary log archive method.

Note:

Skip this step if you only perform full data migration.
Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click Data Migration.
3. At the top of the Migration Tasks page, select the region where the destination RDS instance resides.
4. In the upper-right corner of the page, click Create Migration Task.
5. Configure the information about the source and destination databases for the data migration task.

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Task Name</td>
<td>DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
</tbody>
</table>
## Source Database

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Database</td>
<td>Instance Type</td>
<td>Select an instance type based on where the source database is deployed. The procedure in this topic uses User-Created Database with Public IP Address as an example.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Instance Region</td>
<td>If the instance type is set to User-Created Database with Public IP Address, you do not need to specify the instance region.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Database Type</td>
<td>Select DB2.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Hostname or IP Address</td>
<td>Enter the endpoint that is used to connect to the user-created Db2 database. In this example, enter the public IP address.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Port Number</td>
<td>Enter the service port number of the user-created Db2 database. The default port number is 50000.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Database Name</td>
<td>Enter the name of the source Db2 database.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Database Account</td>
<td>Enter the account for the user-created Db2 database. For more information about permissions required for the account, see Permissions required for database accounts.</td>
</tr>
</tbody>
</table>

**Note:**
If you select other instance types, you must prepare the environments that are required for the source database. For more information, see #unique_143.

**Note:**
If a whitelist is configured for the user-created Db2 database, you must add the CIDR blocks of DTS servers to the whitelist of the user-created Db2 database. You can click Get IP Address Segment of DTS next to Instance Region to obtain the CIDR blocks of DTS servers.
<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Password</td>
<td></td>
<td>Enter the password for the database account.</td>
</tr>
<tr>
<td></td>
<td>Note:</td>
<td>After the source database information is specified, click Test Connectivity next to Database Password to verify whether the specified information is valid. If the specified information is valid, the Passed message appears. If the Failed message appears, click Check in the Failed message. Modify the source database information as prompted.</td>
</tr>
<tr>
<td>Destination</td>
<td>Instance Type</td>
<td>Select RDS Instance.</td>
</tr>
<tr>
<td>Database</td>
<td>Instance Region</td>
<td>Select the region where the destination RDS instance resides.</td>
</tr>
<tr>
<td></td>
<td>RDS Instance ID</td>
<td>Select the ID of the destination RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account of the destination RDS instance. For more information about permissions required for the account, see Permissions required for database accounts.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
<tr>
<td></td>
<td>Note:</td>
<td>After the destination database information is specified, click Test Connectivity next to Database Password to verify whether the specified information is valid. If the specified information is valid, the Passed message appears. If the Failed message appears, click Check in the Failed message. Modify the destination database information as prompted.</td>
</tr>
<tr>
<td>Encryption</td>
<td>Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data migration task. For more information, see Configure SSL encryption for an RDS for MySQL instance.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note:</td>
<td>The Encryption parameter is available only in mainland China and Hong Kong(China).</td>
</tr>
</tbody>
</table>
6. In the lower-right corner of the page, click Set Whitelist and Next.

**Note:**
The CIDR blocks of DTS servers are automatically added to the whitelist of the destination RDS instance. This ensures that DTS servers can connect to the destination RDS instance.

7. Select the migration types and objects to be migrated.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Migration types | • To perform only full data migration, select **Schema Migration** and **Full Data Migration**.  
• To migrate data with minimal downtime, you must select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**.  

**Note:**
If **Incremental Data Migration** is not selected, do not write data into the source database during full data migration. This ensures data consistency between the source and destination databases. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects to be migrated</td>
<td>Select the objects to be migrated in the <strong>Available</strong> section and click icon to move them to the <strong>Selected</strong> section.</td>
</tr>
</tbody>
</table>

**Note:**
- You can select databases, tables, or columns as the objects to be migrated.
- After a Db2 database is migrated to the destination RDS instance, the name of the Db2 database remains unchanged. If you want an object to have a different name after the object is migrated to the destination RDS instance, you can use the object name mapping feature provided by DTS. For more information, see #unique_144.
- If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.

8. Click **Precheck** in the lower-right corner of the page.

**Note:**
- A precheck is performed for a data migration task. A data migration task can be started only if it passes the precheck.
- If the precheck fails, click corresponding to each failed item to view the details.

Fix the problems as instructed and run the precheck again.

9. After the precheck is passed, click **Next**.

10. On the **Confirm Settings** dialog box that appears, specify **Channel Specification** and select the **Data Transmission Service (Pay-As-You-Go) Service Terms**.
11. Click **Buy and Start** to start the data migration task.

- **Full data migration**
  
  Do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the data migration task stops when it is complete.

- **Incremental data migration**
  
  An incremental data migration task does not automatically end. You must manually end the migration task.

  **Note:**

  Select an appropriate time point to manually end the migration task. For example, you can end the migration task during off-peak hours or before you switch your workloads to the destination instance.

  a. When the task progress bar switches to **Incremental Data Migration** and **The migration task is not delayed** appears, stop writing new data to the source database for a few minutes. Then, the progress bar will show the latency of the incremental data migration.

  b. When the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.

12. Switch your workloads to the ApsaraDB RDS for MySQL instance.

### 6.4.6 Migrate data to an RDS MySQL instance by using mysqldump

This topic describes how to migrate data to an RDS MySQL instance by using mysqldump.

mysqldump is easy to use but requires your database to go through a long service
downtime. Therefore, we recommend that you use mysqldump when the data volume is small or a long service downtime is allowed.

**Background information**

ApsaraDB for RDS is fully compatible with native MySQL. Therefore, the process of migrating data from your database to an RDS MySQL instance is similar to the process of migrating data from one MySQL server to another MySQL server.

**Precautions**

The data in the tables after migration is case-insensitive and centrally changed to lowercase letters.

**Prerequisites**

- You have configured a whitelist, obtained a public endpoint, and created databases and accounts for the target RDS instance. For more information, see #unique_152.
- You have purchased an ECS instance.

**Procedure**

1. Create an account used to migrate data in your on-premises database.

   ```
   CREATE USER 'username'@'host' IDENTIFIED BY 'password';
   ```

   Parameter description:

   - **username**: the username of the account.
   - **host**: the name of the host from which you log on to your on-premises database by using the account. As a local user, you can set this parameter to localhost. To log on from any host, you can set this parameter to the wildcard %.
   - **password**: the password of the account.

   For example, if you want to create an account whose username and password are William and Changme123, respectively, and enable the account to log on to your on-premises database from any host, then run the following command:

   ```
   CREATE USER 'William'@'%' IDENTIFIED BY 'Changme123';
   ```

2. Grant permissions to the account in your on-premises database.

   ```
   GRANT SELECT ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION;
   ```
GRANT REPLICATION SLAVE ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION;

Parameter description:

- **privileges**: the permissions of the account, such as SELECT, INSERT, and UPDATE. To grant all permissions to the account, set this parameter to ALL.
- **databasename**: the name of your on-premises database. To grant the permissions for all databases to the account, set this parameter to the wildcard *.
- **tablename**: the name of a table. To grant the permissions for all tables to the account, set this parameter to the wildcard *.
- **username**: the username of the account.
- **host**: the name of the host from which you log on to your on-premises database by using the account. As a local user, you can set this parameter to localhost. To log on from any host, you can set this parameter to the wildcard %.
- **WITH GRANT OPTION**: an optional parameter, which is used to grant the account the permission to run the `GRANT` command.

For example, if you want to grant all permissions for all databases and tables to the account William and allow the account to log on from any host, then run the following command:

```
GRANT ALL ON*. * TO 'William'@'%';
```

3. Use mysqldump to export data from your on-premises database as a data file.

**Note:**

Do not update the data during the data export. This step exports data only, excluding stored procedures, triggers, and functions.

```
mysqldump -h localip -u userName -p --opt --default-character-set=utf8 --hex-blob dbName --skip-triggers --skip-lock-tables > /tmp/dbName.sql
```

Parameter description:

- **localip**: the IP address of the server where your on-premises database is located.
- **userName**: the username of the account used to migrate data from your on-premises database.
- **dbName**: the name of your on-premises database.
- **/tmp/dbName.sql**: the name of the generated backup file.
4. Use mysqldump to export stored procedures, triggers, and functions.

**Note:**
If no stored procedures, triggers, or functions are used in the database, you can skip this step. When exporting stored procedures, triggers, and functions, you must remove the `DEFINER` parameter to guarantee compatibility with RDS.

```bash
mysqldump -h localIp -u userName -p --opt --default-character-set=utf8 --hex-blob dbName -R | sed -e 's/DEFINER\[ \]*=\[ \]*[^\]*\]/\*/' > /tmp/triggerProcedure.sql
```

Parameter description:
- **localIp**: the IP address of the server where your on-premises database is located.
- **userName**: the username of the account used to migrate data from your on-premises database.
- **dbName**: the name of your on-premises database you want to migrate.
- **/tmp/triggerProcedure.sql**: the name of the generated backup file.

5. Upload the data file and stored procedure file to your ECS instance.

In this example, assume that you want to upload the files to the following directories on your ECS instance:

- `/tmp/dbName.sql`
- `/tmp/triggerProcedure.sql`

6. Log on to your ECS instance and import the data file and stored procedure file to the target RDS instance.

```bash
mysql -h intranet4example.mysql.rds.aliyuncs.com -u userName -p dbName < /tmp/dbName.sql
mysql -h intranet4example.mysql.rds.aliyuncs.com -u userName -p dbName < /tmp/triggerProcedure.sql
```

Parameter description:
- **intranet4example.mysql.rds.aliyuncs.com**: the endpoint of the target RDS instance. A private endpoint is used as an example.
- **userName**: the username of the account used to migrate data to the RDS instance.
- **dbName**: the name of your on-premises database you want to import.
- **/tmp/dbName.sql**: the name of the data file you want to import.
- **/tmp/triggerProcedure.sql**: the name of the stored procedure file you want to import.
6.5 Migrate data from a third-party cloud database to ApsaraDB for RDS

6.5.1 Migrate a MySQL database from Google Cloud to Alibaba Cloud

This topic describes how to migrate a MySQL database from Google Cloud to Alibaba Cloud and the corresponding precautions.

Prerequisites

- You have created an Alibaba Cloud RDS MySQL instance.
- You have created an account with read/write privileges.

Limits

- Structure migration does not support migration of events.
- For MySQL databases, DTS reads floating-point values (FLOAT and DOUBLE data types) with \texttt{round(column,precision)}. If the column definition does not specify the precision, the precision is 38 for FLOAT values and 308 for DOUBLE values.
- If the object name mapping function is used for an object, migration of objects relying on the object may fail.
- For incremental migration, you must enable binlog for the source MySQL instance.
- For incremental migration, \texttt{binlog\_format} of the source database must be set to \texttt{ROW}.

\textbf{Note:}
You can modify parameters of Google Cloud databases by choosing \textbf{Instance details} > \textbf{Configuration} > \textbf{Edit configuration} > \textbf{Add database flags}.

- For incremental migration, if the source database version is MySQL 5.6 or later, \texttt{binlog\_row\_image} must be set to \texttt{FULL}.
- For incremental migration, if the source instance has binlog file ID disorder caused by cross-host migration, the incremental migration may have data loss.

Precautions

DTS automatically attempts to recover abnormal tasks of the past seven days. This may cause the new data in the target instance to be overwritten by the source database data. Therefore, you must revoke the write permission of the DTS account that is used to access the target instance by running the revoke command.
Procedure

1. Log on to your database instance on Google Cloud. On the Instance details page, view Public IP address.

Note:
If a public IP address is not enabled, perform related settings by going to Configuration > Edit configuration > Set connectivity.
2. Choose **Configuration > Edit configuration > Set connectivity > Add network**, and then add the IP address of the region of the source database instance obtained from DTS.

3. Log on to the **DTS console**.

4. In the left-side navigation pane, click **Data Migration**. In the right pane, click **Create Migration Task** in the upper-right corner.
5. Enter information about the source and target databases. The following table describes the parameters.

<table>
<thead>
<tr>
<th>Database type</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source database (on Google Cloud)</strong></td>
<td>Instance Type</td>
<td>Type of the instance in the source database. Select On-premises Databases.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>If you have configured access control for your instance, you must allow the specified Internet IP segment of the region to access the instance before configuring a migration task.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> You can click <strong>Get DTS IP</strong> to view and copy the IP segment of the region.</td>
</tr>
<tr>
<td></td>
<td>Database Engine</td>
<td>Source database type. Select MySQL.</td>
</tr>
<tr>
<td></td>
<td>Host Name or IP Address</td>
<td>Public IP address of the database</td>
</tr>
<tr>
<td></td>
<td>Port</td>
<td>Default port 3306</td>
</tr>
<tr>
<td></td>
<td>Database account</td>
<td>Default superuser account root</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Password of the root account</td>
</tr>
<tr>
<td><strong>Target database (on Alibaba Cloud)</strong></td>
<td>Instance Type</td>
<td>Type of the instance in the target database. Select RDS Instance.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>Region of the target instance</td>
</tr>
<tr>
<td></td>
<td>RDS Instance ID</td>
<td>ID of the instance in the selected region. Select the ID of the target instance.</td>
</tr>
<tr>
<td></td>
<td>Database account</td>
<td>An account with read and write permissions under the target instance</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Account password</td>
</tr>
</tbody>
</table>
### Database type | Parameter | Description
--- | --- | ---
 | Connection method | Select **Non-encrypted connection** or **SSL secure connection**. The latter greatly increases CPU consumption. | 

6. Click **Test the Connection** and confirm that the test results for both the source and target databases are Test passed.  
7. Click **Authorize Whitelist and Enter into Next Step**.  
8. Select the migration type. In the **Migration objects** area, select the target database and click **```** to add the database to the **Selected objects** area.

**Note:**
To maintain data consistency before and after migration, we recommend that you migrate the structure, full data, and incremental data.

9. Click Pre-check and wait until the pre-check ends.

Note:
1. Data migration only copies data and structure from source database to target database, it has no influence on source database.
2. DDL operations are not allowed during the process of data migration. Otherwise, the migration task may fail.
If the check fails, you can rectify faults according to error items and restart the task.

10. Click **Next**. In the **Confirm Purchase Configuration** dialog box, read and select **Service Terms of Data Transmission (Pay-As-You-Go)**, then click **Buy and Start Now**.

**Note:**
Currently, structure migration and full migration are free of charge, while incremental migration is charged by the hour according to link specifications.

11. Wait until the migration task is completed.
6.5.2 Migrate data from an Amazon RDS for MySQL database to an ApsaraDB RDS for MySQL database

This topic describes how to migrate data from an Amazon RDS for MySQL database to an ApsaraDB RDS for MySQL database by using Data Transmission Service (DTS). DTS supports schema migration, full data migration, and incremental data migration. When configuring a data migration task, you can select all of the supported migration types to ensure service continuity.

Prerequisites

- The **Public accessibility** option of Amazon RDS for MySQL is set to **Yes**. This ensures that DTS can access Amazon RDS for MySQL over the Internet.
- An ApsaraDB RDS for MySQL instance is created. For more information, see [Create an RDS for MySQL instance](#).
- The available storage space of the destination ApsaraDB RDS for MySQL database is larger than the total space of the data in the Amazon RDS for MySQL database.

Precautions

- DTS uses read and write resources of the source and destination databases during full data migration. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large, database services may become unavailable. For example, DTS occupies a large amount of read and write resources in the following cases: a large number of slow SQL queries are performed on the source database, the tables have no primary keys, or a deadlock occurs in the destination database. Before you migrate data, evaluate the performance of the source and destination databases. We recommend that you migrate data during off-peak hours. For example, you can migrate data when the CPU usage of the source and destination databases is less than 30%.
- The source database must have PRIMARY KEY or UNIQUE constraints and all fields must be unique. Otherwise, duplicate data may exist in the destination database.
- DTS uses the `ROUND(COLUMN, PRECISION)` function to retrieve values from columns of the float or double data type. If the precision is not specified, DTS sets the precision for the float data type to 38 digits and the precision for the double data type to 308 digits. You must check whether the precision settings meet your business requirements.
- DTS automatically creates a destination database in the ApsaraDB RDS for MySQL instance. However, if the name of the source database is invalid, you must manually
create a database in the ApsaraDB RDS for MySQL instance before you configure the data migration task.

**Note:**
For more information about how to create a database and the database naming conventions, see [Create databases and accounts](#).

- DTS automatically resumes a failed data migration task. Before you switch your workloads to the destination database, stop or release the data migration task. Otherwise, the data in the source database will overwrite the data in the destination database after the task is resumed.

### Billing

<table>
<thead>
<tr>
<th>Migration type</th>
<th>Instance configurations</th>
<th>Internet traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema migration and full data migration</td>
<td>Free of charge.</td>
<td>Charged only when data is migrated from Alibaba Cloud over the Internet.</td>
</tr>
<tr>
<td>Incremental data migration</td>
<td>Charged. For more information, see #unique_141.</td>
<td>For more information, see #unique_141.</td>
</tr>
</tbody>
</table>

### Migration types

- **Schema migration**

  DTS migrates the schemas of the required objects to the destination instance. DTS supports schema migration for the following types of objects: table, view, trigger, stored procedure, and function. DTS does not support schema migration for events.

**Note:**
- During schema migration, DTS changes the value of the SECURITY attribute from DEFINER to INVOKER for views, stored procedures, and functions.
- DTS does not migrate user information. Before a user can call views, stored procedures, and functions of the destination database, you must grant the read/write permissions to the user.
• Full data migration

DTS migrates historical data of the required objects from the source Amazon RDS for MySQL database to the destination ApsaraDB RDS for MySQL database.

**Note:**
- During full data migration, concurrent INSERT operations cause segments in the tables of the destination instance. After full data migration is complete, the tablespace of the destination instance is larger than that of the source instance.
- To ensure data consistency, do not write new data into the Amazon RDS for MySQL instance during full data migration.

• Incremental data migration

After full data migration is complete, DTS retrieves binary log files from the source Amazon RDS for MySQL database. Then, DTS synchronizes incremental data from the source Amazon RDS for MySQL database to the destination ApsaraDB RDS for MySQL database. Incremental data migration helps you ensure service continuity when you migrate data between MySQL databases.

### Permissions required for database accounts

<table>
<thead>
<tr>
<th>Database</th>
<th>Schema migration</th>
<th>Full data migration</th>
<th>Incremental data migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon RDS for MySQL database</td>
<td>The SELECT</td>
<td>The SELECT</td>
<td>The SELECT, REPLICATION SLAVE, and REPLICATION CLIENT permissions</td>
</tr>
<tr>
<td></td>
<td>permission</td>
<td>permission</td>
<td></td>
</tr>
<tr>
<td>ApsaraDB RDS for MySQL database</td>
<td>The read/write</td>
<td>The read/write</td>
<td>The read/write permissions</td>
</tr>
<tr>
<td></td>
<td>permissions</td>
<td>permissions</td>
<td></td>
</tr>
</tbody>
</table>

For more information about how to create and authorize a database account, see the following topics:
- #unique_142 for an Amazon RDS for MySQL database
- Create an account for an RDS for MySQL instance and Change the permissions of an account for an RDS for MySQL instance for an ApsaraDB RDS for MySQL database

### Preparations before data migration

1. Log on to the Amazon RDS Management Console.
2. Go to the **Basic information** page of the source Amazon RDS for MySQL instance.

3. In the **Security group rules** section, click the name of the security group corresponding to the existing inbound rule.

4. On the **Security groups** page, click the Inbound tab in the Security group section. On the Inbound tab, click **Edit** to add CIDR blocks of DTS servers in the corresponding region to the inbound rule. For more information, see #unique_154.

**Note:**

- You only need to add the CIDR blocks of DTS servers that are located in the same region as the destination database. For example, the source database is located in Singapore and the destination database is located in Hangzhou. You only need to add the CIDR blocks of DTS servers that are located in the China (Hangzhou) region.
5. Log on to the Amazon RDS for MySQL database and specify the number of hours to retain binary log files. Skip this step if you do not need to perform incremental data migration.

```sql
call mysql.rds_set_configuration('binlog retention hours', 24);
```

**Note:**
- The preceding command sets the retention period of binary log files to 24 hours. The maximum value is 168 hours (7 days).
- The binary logging feature of Amazon RDS for MySQL must be enabled and the value of the binlog_format parameter must be set to row. If the version of MySQL is 5.6 or later, the value of the binlog_row_image parameter must be set to full.

**Procedure**

1. Log on to the DTS console.

2. In the left-side navigation pane, click **Data Migration**.

3. At the top of **Migration Tasks** the page, select the region where the destination cluster resides.

4. In the upper-right corner of the page, click **Create Migration Task**.
5. Configure the information about the source and destination databases for the data migration task.

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Task Name</td>
<td>DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Instance Type</td>
<td>Select <strong>User-Created Database with Public IP Address</strong>.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Instance Region</td>
<td>If the instance type is set to <strong>User-Created Database with Public IP Address</strong>, you do not need to specify the <strong>instance region</strong>.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Database Type</td>
<td>Select <strong>MySQL</strong>.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Hostname or IP Address</td>
<td>Enter the endpoint that is used to connect to the Amazon RDS for MySQL database.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can obtain the endpoint on the <strong>Basic information</strong> page of the source Amazon RDS for MySQL instance.</td>
<td></td>
</tr>
<tr>
<td>Port Number</td>
<td>Enter the service port number of the Amazon RDS for MySQL database. The default port number is 3306.</td>
<td></td>
</tr>
<tr>
<td>Database Account</td>
<td>Enter the account for the Amazon RDS for MySQL database. For more information about permissions required for the account, see Permissions required for database accounts.</td>
<td></td>
</tr>
<tr>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> After the source database information is specified, click <strong>Test Connectivity</strong> next to <strong>Database Password</strong> to verify whether the specified information is valid. If the specified information is valid, the <strong>Passed</strong> message appears. If the <strong>Failed</strong> message appears, click <strong>Check</strong> in the <strong>Failed</strong> message. Modify the source database information as prompted.</td>
<td></td>
</tr>
<tr>
<td>Destination Database</td>
<td>Select <strong>RDS Instance</strong>.</td>
<td></td>
</tr>
<tr>
<td>Instance Region</td>
<td>Select the region where the ApsaraDB RDS for MySQL instance resides.</td>
<td></td>
</tr>
<tr>
<td>RDS Instance ID</td>
<td>Select the ID of the ApsaraDB RDS for MySQL instance.</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account of the ApsaraDB RDS for MySQL instance. For more information about permissions required for the account, see Permissions required for database accounts.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> After the destination database information is specified, click Test Connectivity next to Database Password to verify whether the specified information is valid. If the specified information is valid, the Passed message appears. If the Failed message appears, click Check in the Failed message. Modify the destination database information as prompted.</td>
</tr>
<tr>
<td>Encryption</td>
<td>Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data migration task. For more information, see Configure SSL encryption for an RDS for MySQL instance.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The Encryption parameter is available only in mainland China and Hong Kong(China).</td>
</tr>
</tbody>
</table>

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

**Note:** The CIDR blocks of DTS servers are automatically added to the whitelist of the ApsaraDB for RDS instance. This ensures that DTS servers can connect to the ApsaraDB for RDS instance.
7. Select the migration types and objects to be migrated.

- **Parameter**
  - **Migration types**
    - To perform only full data migration, select **Schema Migration** and **Full Data Migration**.
    - To migrate data with minimal downtime, you must select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**.

  **Note:**
  - If **Incremental Data Migration** is not selected, do not write data into the source database during full data migration. This ensures data consistency between the source and destination databases.
  - During **Schema Migration** and **Full Data Migration**, do not perform DDL operations on the objects to be migrated. Otherwise, the objects may fail to be migrated.
### Objects to be migrated

In the **Available** section, select the objects to be migrated and click the icon to add the objects to the **Selected** section.

**Note:**
- You can select databases, tables, or columns as the objects to be migrated.
- After an object is migrated to the destination instance, the name of the object remains unchanged. If you want an object to have a different name after the object is migrated to the ApsaraDB RDS for MySQL instance, you can use the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_144.
- If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.

8. Click **Precheck** on the lower right of the page.

**Note:**
- A precheck is performed for a data migration task. A data migration task can be started only if it passes the precheck.
- If the precheck fails, click icon corresponding to each failed item to view the details. Fix the problems as instructed and run the precheck again.

9. After the precheck is passed, click **Next**.

10. On the **Confirm Settings** dialog box that appears, specify **Channel Specification** and select the **Data Transmission Service (Pay-As-You-Go) Service Terms**.
11. Click **Buy and Start** to start the data migration task.

- Schema migration and full data migration
  
  Do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the data migration task stops when it is complete.

- Schema migration, full data migration, and incremental data migration
  
  An incremental data migration task does not automatically end. You must manually end the migration task.

**Note:**

Select an appropriate time point to manually end the migration task. For example, you can end the migration task during off-peak hours or before you switch your workloads to the destination cluster.

a. When the task progress bar switches to **Incremental Data Migration** and the message **The migration task is not delayed** appears, stop writing new data to the source database for a few minutes. Then, the progress bar will show the latency of the incremental data migration.

b. When the status of incremental data migration changes to **The migration task is not delayed**, manually stop the migration task.

12. Switch your workloads to the ApsaraDB RDS for MySQL instance.

### 6.6 Migrate data between ApsaraDB for RDS instances

You can seamlessly migrate data between ApsaraDB for RDS instances.

**Notes**

- The source and destination RDS instances must use the same database engines. DTS does not support data migration between RDS instances that use different database engines.
• If the source database does not have primary keys or UNIQUE constraints, and fields are not required to be unique, duplicate data may exist in the destination database.

• To ensure data consistency, do not write new data into the source RDS instance during full data migration.

• DTS automatically resumes a failed data migration task. Before switching your workloads to the destination instance, you must stop or release the data migration task. Otherwise, the data from the source database will overwrite the data in the destination instance after the task is resumed.

Billing

<table>
<thead>
<tr>
<th>Migration type</th>
<th>Migration channel fee</th>
<th>Public network traffic fee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema migration or full data migration</td>
<td>Free of charge</td>
<td>Migrating data from Alibaba Cloud over the Internet incurs fees. For more information, see #unique_141.</td>
</tr>
<tr>
<td>Incremental data migration</td>
<td>Billed. For more information, see #unique_141.</td>
<td></td>
</tr>
</tbody>
</table>

Migration types

• Schema migration

  DTS migrates the schemas of the required objects to the destination instance.

• Full data migration

  DTS migrates historical data of the required objects from the source RDS instance to the destination RDS instance.

• Incremental data migration

  After full data migration is complete, DTS synchronizes incremental data from the source RDS instance to the destination RDS instance. Incremental data migration helps you ensure service continuity when you migrate data between RDS instances.

Permissions required for database accounts

<table>
<thead>
<tr>
<th>Database</th>
<th>Schema migration</th>
<th>Full data migration</th>
<th>Incremental data migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source RDS instance</td>
<td>The read/write permissions</td>
<td>The read/write permissions</td>
<td>The read/write permissions</td>
</tr>
<tr>
<td>Database</td>
<td>Schema migration</td>
<td>Full data migration</td>
<td>Incremental data migration</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>------------------</td>
<td>---------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Destination RDS instance</td>
<td>The read/write permissions</td>
<td>The read/write permissions</td>
<td>The read/write permissions</td>
</tr>
</tbody>
</table>

**Procedure**

1. Log on to the [DTS console](#).
2. In the left-side navigation pane, click **Data Migration**.
3. At the top of the **Migration Tasks** page, select the region where the destination RDS instance resides.
4. In the upper-right corner of the page, click **Create Migration Task**.
5. Configure the information about the source and destination databases for the data migration task.

### Section: Task Name

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
</tbody>
</table>

### Source Database

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Type</td>
<td>Select <strong>RDS Instance</strong>.</td>
</tr>
<tr>
<td>Instance Region</td>
<td>Select the region where the destination RDS instance resides.</td>
</tr>
<tr>
<td>RDS Instance ID</td>
<td>Select the ID of the source RDS instance.</td>
</tr>
</tbody>
</table>

**Note:**

The source and destination RDS instances can be the same or different. You can use DTS to migrate data within an RDS instance or between two RDS instances.

### Database Name

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter the name of the source database in the ApsaraDB RDS for PostgreSQL instance.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

This parameter is required only if the database engine of the RDS instance is **PostgreSQL**.
<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Account</td>
<td>Database Account</td>
<td>Enter the database account of the source RDS instance. For more information about permissions required for the account, see Permissions required for database accounts.</td>
</tr>
<tr>
<td>Database Password</td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>After the source database parameters are specified, click Test Connectivity next to the Database Password parameter to verify whether the specified parameters are correct. If the source database parameters are correct, the Test Passed message is displayed, if the Test Failed message is displayed, click Diagnose in the Test Failed message. Modify the source database parameters as prompted.</td>
</tr>
<tr>
<td>Encryption</td>
<td>Encryption</td>
<td>Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data migration task. For more information, see Configure SSL encryption for an RDS for MySQL instance.</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>This parameter is required only if the database engine of the RDS instance is MySQL. The Encryption parameter is available only in mainland China and Hong Kong(China).</td>
</tr>
<tr>
<td>Destination Database</td>
<td>Instance Type</td>
<td>Select RDS Instance.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>Select the region where the destination RDS instance resides.</td>
</tr>
<tr>
<td></td>
<td>RDS Instance ID</td>
<td>Select the ID of the destination RDS instance.</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>The source and destination RDS instances can be the same or different. You can use DTS to migrate data within an RDS instance or between two RDS instances.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Database Name</td>
<td>Enter the name of the destination database in the ApsaraDB RDS for PostgreSQL instance. The name of the destination database can be different from that of the source database.</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>This parameter is required only if the database engine of the RDS instance is PostgreSQL.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account of the destination RDS instance. For more information about permissions required for the account, see Permissions required for database accounts.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>After the destination database parameters are specified, click Test Connectivity next to the Database Password parameter to verify whether the specified parameters are correct. If the destination database parameters are correct, the Test Passed message is displayed. If the Test Failed message is displayed, click Diagnose in the Test Failed message. Modify the destination database parameters as prompted.</td>
</tr>
<tr>
<td>Encryption</td>
<td></td>
<td>Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data migration task. For more information, see Configure SSL encryption for an RDS for MySQL instance.</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>This parameter is required only if the database engine of the RDS instance is MySQL. The Encryption parameter is available only in mainland China and Hong Kong(China).</td>
</tr>
</tbody>
</table>

6. In the lower-right corner of the page, click Set Whitelist and Next.
The CIDR blocks of DTS servers are automatically added to the whitelist of the source and destination RDS instances. This ensures that DTS servers can connect to the RDS instances.

7. Select the migration types and objects to be migrated.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migration types</td>
<td>Select the migration types based on your requirements. The migration types must be supported by the database engine.</td>
</tr>
<tr>
<td></td>
<td>• To perform only full data migration, select <strong>Schema Migration</strong> and <strong>Full Data Migration</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> To ensure data consistency, do not write new data into the source RDS instance during full data migration.</td>
</tr>
<tr>
<td></td>
<td>• To migrate data with minimal downtime, you must select <strong>Schema Migration</strong>, <strong>Full Data Migration</strong>, and <strong>Incremental Data Migration</strong>.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Objects to be migrated</td>
<td>In the <strong>Available</strong> section, select the objects to be migrated and click the <img src="image" alt="icon" /> to add the objects to the <strong>Selected</strong> section.</td>
</tr>
</tbody>
</table>

**Note:**
- You can select databases, tables, or columns as the objects to be migrated.
- After an object is migrated to the destination RDS instance, the name of the object remains the same as that in the source RDS instance. If the migrated object has a different name in the destination RDS instance, you can use the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_144.
- If you use the object name mapping feature on an object, other objects that are dependent on the object may fail to be migrated.

8. Click **Precheck** in the lower-right corner of the page.

**Note:**
- A precheck is performed for a data migration task. A data migration task can be started only if it passes the precheck.
- If the precheck fails, click ![icon](image) corresponding to each failed item to view the details.

Fix the problems as instructed and run the precheck again.

9. After the precheck is passed, click **Next**.

10. On the **Confirm Settings** dialog box that appears, specify **Channel Specification** and select the **Data Transmission Service (Pay-As-You-Go) Service Terms**.
11. Click **Buy and Start** to start the data migration task.

- **Full data migration**
  
  Do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the data migration task stops when it is complete.

- **Incremental data migration**
  
  An incremental data migration task does not automatically end. You must manually end the migration task.

  **Note:**
  
  Select an appropriate time point to manually end the migration task. For example, you can end the migration task during off-peak hours or before you switch your workloads to the destination instance.

  a. When the task progress bar switches to **Incremental Data Migration** and **The migration task is not delayed** appears, stop writing new data to the source database for a few minutes. Then, the progress bar will show the latency of the incremental data migration.

  b. When the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.

### 6.7 Migrate data from an ApsaraDB RDS for MySQL database to a user-created MySQL database

This topic describes how to migrate data from an ApsaraDB RDS for MySQL database to a user-created MySQL database by using Data Transmission Service (DTS). This is applicable to scenarios such as data analysis and functional test.

**Prerequisites**

- The tables to be migrated from the ApsaraDB RDS for MySQL database contain primary keys or UNIQUE NOT NULL indexes.
• The available storage space of the user-created MySQL database is greater than the total size of the data in the ApsaraDB RDS for MySQL database.

• The version of the user-created MySQL database is the same as that of the ApsaraDB RDS for MySQL database. This ensures compatibility.

Background information

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

• If your ApsaraDB RDS for MySQL database does not have primary key or unique constraints and each field in the database has duplicate values, the data migrated to the destination database may be duplicated.

• Concurrent insertions are performed during full data migration. This results in table fragmentation in the destination instance. After a full data migration task is completed, the tablespace of the destination instance is larger than that of the source instance.

• If a data migration task fails, DTS attempts to resume the task. In this case, before you switch your workloads to the destination database, you must stop or release the task. Otherwise, the data in the source database will overwrite the data in the destination database after the task is resumed.

Limits

• DTS supports schema migration of the following objects: tables, views, triggers, stored procedures, and stored functions.

Note:
During schema migration, the DEFINER mode of views, stored procedures, and stored functions is shifted to the INVOKER mode.
• The information of the source database account cannot be migrated. If you need to use views, stored procedures, and stored functions, you must grant read and write permissions to the destination database account.

Migration types

DTS supports schema migration, full data migration, and incremental data migration. For more information, see #unique_156.

Note:
You can use these three migration types together to migrate data without service interruptions.

Billing

<table>
<thead>
<tr>
<th>Migration type</th>
<th>Migration channel fee</th>
<th>Public network traffic fee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema migration or full data migration</td>
<td>Free of charge</td>
<td>Migrating data from Alibaba Cloud over the Internet incurs fees. For more information, see #unique_141.</td>
</tr>
<tr>
<td>Incremental data migration</td>
<td>Billed. For more information, see #unique_141.</td>
<td></td>
</tr>
</tbody>
</table>

SQL operations that can be synchronized during incremental data migration

<table>
<thead>
<tr>
<th>Operation type</th>
<th>SQL statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DML</td>
<td>INSERT, UPDATE, DELETE, and REPLACE</td>
</tr>
<tr>
<td>DDL</td>
<td>• ALTER TABLE and ALTER VIEW</td>
</tr>
<tr>
<td></td>
<td>• CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW</td>
</tr>
<tr>
<td></td>
<td>• DROP INDEX and DROP TABLE</td>
</tr>
<tr>
<td></td>
<td>• RENAME TABLE</td>
</tr>
<tr>
<td></td>
<td>• TRUNCATE TABLE</td>
</tr>
</tbody>
</table>
Permissions required for database accounts

<table>
<thead>
<tr>
<th>Database</th>
<th>Permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApsaraDB RDS for MySQL database</td>
<td>The read permission for the objects to be migrated</td>
</tr>
<tr>
<td>User-created MySQL database</td>
<td>The read/write permissions for the objects to be migrated</td>
</tr>
</tbody>
</table>

For more information about how to create and authorize a database account, see the following topics:

- Create an account for an RDS for MySQL instance and Change the permissions of an account for an RDS for MySQL instance
- #unique_142 for a user-created MySQL database

Procedure

1. Log on to the DTS console.
2. In the left-side navigation pane, click Data Migration.
3. At the top of Migration Tasks the page, select the region where the destination cluster resides.
4. In the upper-right corner of the page, click Create Migration Task.
5. Configure the information about the source and destination databases for the data migration task.

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Task Name</td>
<td>DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Source Database</td>
<td>Instance Type</td>
<td>Select <strong>RDS Instance</strong>.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>Select the region where the source RDS instance resides.</td>
</tr>
<tr>
<td></td>
<td>RDS Instance ID</td>
<td>Select the ID of the source RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account of the source RDS instance. For more information about permissions required for the account, see <strong>Permissions required for database accounts</strong>.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> After the source database information is specified, click <strong>Test Connectivity</strong> next to <strong>Database Password</strong> to verify whether the specified information is valid. If the specified information is valid, the <strong>Passed</strong> message appears. If the <strong>Failed</strong> message appears, click <strong>Check</strong> in the <strong>Failed</strong> message. Modify the source database information based on the instructions.</td>
</tr>
<tr>
<td></td>
<td>Encryption</td>
<td>Select <strong>Non-encrypted</strong> or <strong>SSL-encrypted</strong>. If you want to select <strong>SSL-encrypted</strong>, you must enable SSL encryption for the RDS instance before configuring the data migration task. For more information, see Configure SSL encryption for an RDS for MySQL instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The <strong>Encryption</strong> parameter is available only in mainland China and Hong Kong(China).</td>
</tr>
<tr>
<td>Destination Database</td>
<td>Instance Type</td>
<td>Select <strong>User-Created Database in ECS Instance</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Instance Region</strong> Select the region where the ECS instance resides.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ECS Instance ID</strong> Select the ID of the ECS instance that is connected to the user-created MySQL database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Database Type</strong> Select <strong>MySQL</strong>.</td>
</tr>
<tr>
<td></td>
<td>Port Number</td>
<td>Enter the service port number of the user-created MySQL database. In this example, enter <strong>3306</strong>.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the account of the user-created MySQL database. For more information about permissions required for the account, see Permissions required for database accounts.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Database</td>
<td>Database Password</td>
<td>Enter the password for the account of the user-created MySQL database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> After the destination database information is specified, click <strong>Test Connectivity</strong> next to <strong>Database Password</strong> to verify whether the specified information is valid. If the specified information is valid, the <strong>Passed</strong> message appears. If the <strong>Failed</strong> message appears, click <strong>Check</strong> in the <strong>Failed</strong> message. Modify the destination database information based on the instructions.</td>
</tr>
<tr>
<td>Encryption</td>
<td>Encryption</td>
<td>Select <strong>Non-encrypted</strong> or <strong>SSL-encrypted</strong>. In this example, <strong>Non-encrypted</strong> is selected.</td>
</tr>
</tbody>
</table>

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

**Note:**
The CIDR blocks of DTS servers are automatically added to the whitelist of the source RDS instance and the inbound rule of the destination ECS instance. This ensures that DTS servers can connect to the source and destination instances.
7. Configure migration types and objects.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
</table>
| Migration types | • To perform only full data migration, select **Schema Migration** and **Full Data Migration**.  
|               | • If you want to migrate data without disruptions to your business, select **Schema Migration**, **Full Data Migration**, and **Incremental Data Migration**. |

**Note:**
If **Incremental Data Migration** is not selected, do not write data into the source database during full data migration. This ensures data consistency between the source and destination databases.
### Data Migration

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects to be migrated</td>
<td>Select the objects to be migrated in the Available section and click icon to move them to the Selected section.</td>
</tr>
</tbody>
</table>

**Note:**
- Objects to be migrated can be databases, tables, or columns.
- The selected objects are not renamed after the migration by default. If you want to rename the objects migrated to the destination instance, you can use the object name mapping feature provided by DTS. For more information, see #unique_144.
- If you use the object name mapping feature for an object, objects that depend on the object may fail to be migrated.

8. Click **Precheck** on the lower right of the page.

**Note:**
- A precheck is performed for a data migration task. A data migration task can be started only if it passes the precheck.
- If the precheck fails, click icon corresponding to each failed item to view the details. Fix the problems as instructed and run the precheck again.

9. After the precheck is passed, click **Next**.

10. On the **Confirm Settings** dialog box that appears, specify **Channel Specification** and select the **Data Transmission Service (Pay-As-You-Go) Service Terms**.

11. Click **Buy and Start** to start the data migration task.

   - **Schema migration and full data migration**
     
     Do not manually stop a migration task. Otherwise, data migrated to the destination database will be incomplete. Wait until the data migration task stops when it is complete.

   - **Schema migration, full data migration, and incremental data migration**
     
     An incremental data migration task does not automatically end. You must manually end the migration task.

   **Note:**
Select an appropriate time point to manually end the migration task. For example, you can end the migration task during off-peak hours or before you switch your workloads to the destination cluster.

**a.** When the task progress bar switches to **Incremental Data Migration** and the message **The migration task is not delayed** appears, stop writing new data to the source database for a few minutes. Then, the progress bar will show the latency of the **incremental data migration**.

**b.** When the status of **incremental data migration** changes to **The migration task is not delayed**, manually stop the migration task.
7 Data Synchronization

7.1 Overview of data synchronization

This topic provides an overview of the data synchronization solutions supported by ApsaraDB RDS for MySQL.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Reference</th>
</tr>
</thead>
</table>
| Synchronize data between ApsaraDB RDS for MySQL instances | • Configure two-way data synchronization between ApsaraDB RDS for MySQL instances  
  • Configure one-way data synchronization between ApsaraDB RDS for MySQL instances  
  • Synchronize data from a user-created MySQL database hosted on ECS to an ApsaraDB RDS for MySQL instance  
  • Synchronize data from a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway to an ApsaraDB RDS for MySQL instance  
  • Synchronize data between ApsaraDB RDS for MySQL instances that belong to different Alibaba Cloud accounts |
| Synchronize data from an ApsaraDB RDS for MySQL instance to other types of databases | Synchronize data from an ApsaraDB RDS MySQL instance to a MaxCompute project |

7.2 Synchronize data from MySQL to MySQL

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

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

Prerequisites

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

**Notes**

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

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

**Notice:**

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

• You cannot synchronize data between ApsaraDB RDS for MySQL instances that reside in Zone A of the China (Hong Kong) region.

• The source and destination ApsaraDB RDS for MySQL instances must have internal endpoints.

• If the source database does not have primary keys or UNIQUE constraints, and fields are not required to be unique, duplicate data may exist in the destination database.

• During initial full data synchronization, concurrent INSERT operations cause fragmentation in the tables of the destination instance. After initial full data synchronization, the tablespace of the destination instance is larger than that of the source instance.

**Supported synchronization topologies**

• One-way one-to-one synchronization

• One-way one-to-many synchronization

• One-way cascade synchronization

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

For more information about synchronization topologies, see #unique_165.

SQL operations that can be synchronized

<table>
<thead>
<tr>
<th>Operation type</th>
<th>SQL statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DML</td>
<td>INSERT, UPDATE, DELETE, and REPLACE</td>
</tr>
<tr>
<td>DDL</td>
<td>• ALTER TABLE and ALTER VIEW</td>
</tr>
<tr>
<td></td>
<td>• CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW</td>
</tr>
<tr>
<td></td>
<td>• DROP INDEX and DROP TABLE</td>
</tr>
<tr>
<td></td>
<td>• RENAME TABLE</td>
</tr>
<tr>
<td></td>
<td>• TRUNCATE TABLE</td>
</tr>
</tbody>
</table>

Limits

• Incompatibility of triggers

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

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

• Limits on RENAME TABLE operations

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

1. Purchase an instance for one-way data synchronization. For more information, see #unique_166/unique_166_Connect_42_section_39h_fto_gdl.

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

2. Log on to the DTS console.

3. In the left-side navigation pane, click Data Synchronization.

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

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Synchronization Task Name</td>
<td>DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Source Instance Details</td>
<td>Instance Type</td>
<td>Select RDS Instance.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the source instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>Instance ID</td>
<td>Select the ID of the source RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account for the source RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
</tbody>
</table>

**Note:**
If the database engine of the source RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account or database password.
### Encryption
Select **Non-encrypted** or **SSL-encrypted**. If you want to select **SSL-encrypted**, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see [Configure SSL encryption for an RDS for MySQL instance](#).

**Note:**
The Encryption parameter is available only in mainland China and Hong Kong(China).

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption</td>
<td>Encryption</td>
<td>Select <strong>Non-encrypted</strong> or <strong>SSL-encrypted</strong>. If you want to select <strong>SSL-encrypted</strong>, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see <a href="#">Configure SSL encryption for an RDS for MySQL instance</a>.</td>
</tr>
<tr>
<td>Destination Instance Details</td>
<td>Instance Type</td>
<td>Select <strong>RDS Instance</strong>.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the destination instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>Instance ID</td>
<td>Select the ID of the destination RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account for the destination RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
<tr>
<td></td>
<td>Encryption</td>
<td>Select <strong>Non-encrypted</strong> or <strong>SSL-encrypted</strong>. If you want to select <strong>SSL-encrypted</strong>, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see <a href="#">Configure SSL encryption for an RDS for MySQL instance</a>.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Processing Mode</th>
<th>Description</th>
</tr>
</thead>
</table>
| In Existed Target Table | - **Pre-check and Intercept**: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started.  

**Note:**  
If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see #unique_167.  

- **Ignore**: skips the precheck for identical table names in the source and destination databases.  

**Warning:**  
If you select **Ignore**, data consistency is not guaranteed and your business may be exposed to potential risks.  
- If the source and destination databases have the same schema, and the primary key of a record in the destination database is the same as that in the source database, the record remains unchanged during initial data synchronization. However, the record is overwritten during incremental data synchronization.  
- If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects to be synchronized</td>
<td>Select objects from the <strong>Available</strong> section and click the icon to move the objects to the <strong>Selected</strong> section. You can select tables and databases as the objects to be synchronized.</td>
</tr>
</tbody>
</table>

**Note:**
- If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.
- After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination instance by using the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_167.

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

10. Configure initial synchronization.

   ![Initial Synchronization Options](image)

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

11. In the lower-right corner of the page, click **Precheck**.
7.2.2 Configure two-way data synchronization between ApsaraDB RDS for MySQL instances

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

**Prerequisites**

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

**Precautions**

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

**Supported synchronization topology**

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

**Supported data sources**

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

<table>
<thead>
<tr>
<th>Source database</th>
<th>Destination database</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ApsaraDB RDS for MySQL instance</td>
<td>• ApsaraDB RDS for MySQL instance</td>
</tr>
<tr>
<td>• User-created database hosted on ECS</td>
<td>• User-created database hosted on ECS</td>
</tr>
<tr>
<td>• User-created database connected over Express Connect, VPN Gateway, or Smart</td>
<td>• User-created database connected over Express</td>
</tr>
<tr>
<td>Access Gateway</td>
<td>Connect, VPN Gateway, or Smart Access Gateway</td>
</tr>
<tr>
<td>• User-created database connected over a database gateway</td>
<td>• User-created database connected over a database</td>
</tr>
<tr>
<td>• Self built database accessed through Cloud Enterprise Network(CEN)</td>
<td>gateway</td>
</tr>
<tr>
<td></td>
<td>• Self built database accessed through Cloud</td>
</tr>
<tr>
<td></td>
<td>Enterprise Network(CEN)</td>
</tr>
</tbody>
</table>
### Operations that can be synchronized

<table>
<thead>
<tr>
<th>Operation type</th>
<th>SQL statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DML</td>
<td>INSERT, UPDATE, DELETE, and REPLACE</td>
</tr>
</tbody>
</table>
| DDL           | • ALTER TABLE and ALTER VIEW  
|               | • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW  
|               | • DROP INDEX and DROP TABLE  
|               | • RENAME TABLE  
|               | • TRUNCATE TABLE |

### Conflict detection

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

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

- **Uniqueness conflicts caused by INSERT operations**

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

- **Inconsistent records caused by UPDATE operations**

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

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

**Notice:**

• During two-way synchronization, the system time of the source and destination instances may be different. Synchronization latency may occur. For these reasons, DTS does not guarantee that the conflict detection mechanism can prevent all data conflicts. To perform two-way synchronization, make sure that data records with the same primary key, business primary key, or unique key are updated only on one of the synchronization nodes.

• DTS provides conflict resolution policies to prevent conflicts that may occur during data synchronization. You can select a conflict resolution policy when you configure a two-way data synchronization task.

**Limits**

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

**Notice:**

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

• Incompatibility with triggers

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

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

- **Limits on RENAME TABLE operations**

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

- **Limits on DDL synchronization direction**

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

**Procedure**

1. Purchase an instance for two-way data synchronization. For more information, see #unique_166/unique_166_Connect_42_section_39h_fto_gdl.

   ![Notice](image)

   **Notice:**

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

2. Log on to the DTS console.

3. In the left-side navigation pane, click Data Synchronization.

4. At the top of the Synchronization Tasks page, select the region where the destination instance resides.
5. Find the data synchronization instance and click **Configure Synchronization Channel** in the Actions column of the first data synchronization task.

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

6. Configure the source and destination instances.

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Synchronization Task Name</td>
<td>DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Source Instance Details</strong></td>
<td>Instance Type</td>
<td>Select <strong>RDS Instance</strong>.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>Instance ID</td>
<td>Select the ID of the source RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account of the source RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Notice:</td>
<td>If the database engine of the source RDS instance is <strong>MySQL 5.5</strong> or <strong>MySQL 5.6</strong>, you do not need to configure the database account or database password.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the source database account.</td>
</tr>
<tr>
<td></td>
<td>Encryption</td>
<td>Select <strong>Non-encrypted</strong> or <strong>SSL-encrypted</strong>. If you want to select <strong>SSL-encrypted</strong>, you must enable SSL encryption for the RDS instance before you configure the data synchronization task. For more information, see <a href="#">Configure SSL encryption for an RDS for MySQL instance</a>.</td>
</tr>
<tr>
<td></td>
<td>Notice:</td>
<td>The <strong>Encryption</strong> parameter is available only for regions in mainland China and the Hong Kong (China) region.</td>
</tr>
<tr>
<td><strong>Destination Instance Details</strong></td>
<td>Instance Type</td>
<td>Select <strong>RDS Instance</strong>.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>Instance ID</td>
<td>Select the ID of the destination RDS instance.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account of the destination RDS instance. <strong>Notice:</strong> If the database engine of the destination RDS instance is <strong>MySQL 5.5</strong> or <strong>MySQL 5.6</strong>, you do not need to configure the database account or database password.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the destination database account.</td>
</tr>
<tr>
<td></td>
<td>Encryption</td>
<td>Select <strong>Non-encrypted</strong> or <strong>SSL-encrypted</strong>. If you want to select <strong>SSL-encrypted</strong>, you must enable SSL encryption for the RDS instance before you configure the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance. <strong>Notice:</strong> The Encryption parameter is available only for regions in mainland China and the Hong Kong (China) region.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization policy</td>
<td>Exclude DDL Statements</td>
<td>• To exclude DDL operations, select <strong>Yes</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• To include DDL operations, select <strong>No</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Notice:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>After you select No, the table does not support synchronizing DDL operations in the opposite direction.</td>
</tr>
<tr>
<td></td>
<td>DML Statements for Synchronization</td>
<td>Select the type of DML operations to be synchronized. By default, the <strong>INSERT</strong>, <strong>UPDATE</strong>, and <strong>DELETE</strong> operations are selected. You can select the DML operation types based on your business requirements.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Conflict Resolution Policy</td>
<td>Select the resolution policy in case of a synchronization conflict. By default, <strong>TaskFailed</strong> is selected. You can select a conflict resolution policy based on your business requirements.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>TaskFailed</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default conflict resolution policy. If a conflict occurs during data synchronization, the synchronization task reports an error and exits the process. The task enters a failed state and you must manually resolve the conflict.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Ignore</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If a conflict occurs during data synchronization, the synchronization task ignores the current statement and continues the process. The conflicting records in the destination database are used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Overwrite</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If a conflict occurs during data synchronization, the conflicting records in the destination database are overwritten.</td>
</tr>
</tbody>
</table>
### Processing Mode In Existed Target Table

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pre-check and Intercept</td>
<td>checks whether the destination database contains tables that have the same names as tables in the source database. If the source and destination databases do not contain identical table names, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started.</td>
</tr>
<tr>
<td></td>
<td>Ignore</td>
<td>skips the precheck for identical table names in the source and destination databases.</td>
</tr>
</tbody>
</table>

**Notice:**
You can change the names of the tables to be synchronized by using the object name mapping feature. You can use this feature if the source and destination databases contain identical table names and tables in the destination database cannot be deleted or renamed. For more information, see #unique_167.

**Warning:**
If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks.

- DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization.
- If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails.
### Section Parameter Description

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects to be synchronized</td>
<td>N/A</td>
<td>Select objects (tables or a database) from the Available section and click the icon to move the objects to the Selected section.</td>
</tr>
</tbody>
</table>

**Notice:**

- If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. If you select a table as the object to be synchronized, only schema changes in this table are synchronized to the destination database.
- After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the names of the objects that are synchronized to the destination database by using the object name mapping feature. For more information about how to use this feature, see #unique_167.

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

10. **Configure initial synchronization.**

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

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

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

**Notice:**

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

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

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

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

14. Find the second data synchronization task and click **Configure Synchronization Channel** in the Actions column. Configure data synchronization by following step 5 to step 12.
15. After the second data synchronization task is configured, wait until both tasks are in the **Synchronizing** state. The two-way data synchronization task is configured.

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

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

#### Prerequisites

- The destination RDS instance is created. For more information, see [Create an RDS instance](#).
- The version of the user-created MySQL database is 5.1, 5.5, 5.6, 5.7, or 8.0.
- The user-created MySQL database is connected to Alibaba Cloud VPC over Express Connect, VPN Gateway, or Smart Access Gateway. For more information, see #unique_147.

#### Note:

DTS is allowed to access the VPC to which the user-created MySQL database belongs. For more information, see #unique_150.

#### Notes

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

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

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

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

**Supported synchronization topologies**

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

For more information about synchronization topologies, see #unique_165.

**SQL operations that can be synchronized**

<table>
<thead>
<tr>
<th>Operation type</th>
<th>SQL statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DML</td>
<td>INSERT, UPDATE, DELETE, and REPLACE</td>
</tr>
</tbody>
</table>
### Operation type

<table>
<thead>
<tr>
<th>Operation type</th>
<th>SQL statements</th>
</tr>
</thead>
</table>
| DDL            | • ALTER TABLE and ALTER VIEW  
|                | • CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW  
|                | • DROP INDEX and DROP TABLE  
|                | • RENAME TABLE  
|                | • TRUNCATE TABLE |

### Limits

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

  To avoid this situation, you must delete the trigger that is synchronized to the destination instance and select Table B as the object to be synchronized. For more information, see [Configure synchronization when triggers exist](#unique_142).

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

### Preparations

Before configuring the data synchronization task, you must create a database account and configure binary logging. For more information, see [#unique_142](#unique_142).

### Procedure

1. Purchase a data synchronization instance. For more information, see [#unique_166](#unique_166).
Select **MySQL** for both the source instance and the destination instance. Select **One-Way Synchronization** as the synchronization topology.

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

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

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the account for the user-created MySQL database. The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
<tr>
<td>Destination Instance Details</td>
<td>Instance Type</td>
<td>Select RDS Instance.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the destination instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>Instance ID</td>
<td>Select the ID of the destination RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account for the destination RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
<tr>
<td></td>
<td>Encryption</td>
<td>Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: The Encryption parameter is available only in mainland China and Hong Kong(China).</td>
</tr>
</tbody>
</table>

7. In the lower-right corner of the page, click **Set Whitelist and Next**.
8. Configure the synchronization policy and objects.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Processing Mode In Existed Target Table | • **Pre-check and Intercept**: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started.  

  **Note:**  
  If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see #unique_167.  

  • **Ignore**: skips the precheck for identical table names in the source and destination databases.  

  **Warning:**  
  If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks.  
  - If the source and destination databases have the same schema, and the primary key of a record in the destination database is the same as that in the source database, the record remains unchanged during initial data synchronization. However, the record is overwritten during incremental data synchronization.  
  - If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. |
### Parameter Description

| Objects to be synchronized | Select objects from the **Available** section and click the icon to move the objects to the **Selected** section. You can select tables and databases as the objects to be synchronized. |

**Note:**
- If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.
- After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination instance by using the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_167.

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

10. Configure initial synchronization.

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

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

**Note:**
- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
7.2.4 Synchronize data between ApsaraDB RDS for MySQL instances that belong to different Alibaba Cloud accounts

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

**Prerequisites**

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

**Notes**

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

**Notice:**

To avoid synchronization failure, you can use Data Management (DMS) to perform online DDL schema changes during data synchronization. For more information, see [Change the table schema without locking](#).
• You cannot synchronize data between ApsaraDB RDS for MySQL instances that reside in Zone A of the China (Hong Kong) region.

• The source and destination ApsaraDB RDS for MySQL instances must have an internal endpoint.

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

• During initial full data synchronization, concurrent INSERT operations cause segments in the tables of the destination instance. After initial full data synchronization, the tablespace of the destination instance is larger than that of the source instance.

Supported synchronization topologies

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

For more information about synchronization topologies, see #unique_165.

SQL operations that can be synchronized

Limits

• Incompatibility of triggers

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

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

• Limits on RENAME TABLE operations

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

Preparations

Set the Alibaba Cloud account that owns the destination RDS instance as a trusted account. This allows DTS to access the cloud resources of the Alibaba Cloud account that owns the source RDS instance. For more information, see #unique_169.

Procedure

1. #unique_166/unique_166_Connect_42_section_39h_fto_gdl.

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

2. Log on to the DTS console with the Alibaba Cloud account that owns the destination RDS instance.

3. In the left-side navigation pane, click Data Synchronization.

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

   ![Data Synchronization Tasks](image)

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization Task Name</td>
<td>N/A</td>
<td>DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Source Instance Details</td>
<td>Instance Type</td>
<td>Select <strong>RDS Instance</strong>.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the source instance that you select when purchasing the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>Apsara Stack Tenant Account ID of RDS Instance</td>
<td>Enter the ID of the Alibaba Cloud account that owns the source RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Note:</td>
<td><strong>Note:</strong> Before you configure this parameter, click <strong>RDS Instances of Other Apsara Stack Accounts</strong> in the <strong>Source Instance Details</strong> section.</td>
</tr>
</tbody>
</table>
### Section | Parameter | Description
--- | --- | ---
Role Name | Enter the name of the RAM role that you configure in Preparations. | |
RDS Instance ID | Select the ID of the source RDS instance. | |
**Destination Instance Details** | Instance Type | Select RDS Instance. |
| Instance Region | The region of the destination instance that you select when purchasing the data synchronization instance. You cannot change the value of this parameter. | |
| Instance ID | Select the ID of the destination RDS instance. | |
| Database Account | Enter the database account of the destination RDS instance. |

**Note:**
If the database type of the destination RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account and database password.

| | Database Password | Enter the password for the database account. |
| Encryption | Select Non-encrypted or SSL-encrypted. If you want to select SSL-encrypted, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see Configure SSL encryption for an RDS for MySQL instance.

**Notice:**
The Encryption parameter is available only for regions in mainland China and the Hong Kong (China) region.

7. In the lower-right corner of the page, click Set Whitelist and Next.
8. Configure the synchronization policy and objects.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Processing Mode In Existed Target Table | • **Pre-check and Intercept**: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started.  

**Note:** If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see #unique_167.  

• **Ignore**: skips the precheck for identical table names in the source and destination databases.  

**Warning:** If you select **Ignore**, data consistency is not guaranteed and your business may be exposed to potential risks.  

- If the source and destination databases have the same schema, and the primary key of a record in the destination database is the same as that in the source database, the record remains unchanged during initial data synchronization. However, the record is overwritten during incremental data synchronization.  

- If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects to be synchronized</td>
<td>Select objects from the <strong>Available</strong> section and click the <img src="image" alt="icon" /> icon to move the objects to the <strong>Selected</strong> section. You can select tables and databases as the objects to be synchronized.</td>
</tr>
</tbody>
</table>

**Note:**
- If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.
- After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination instance by using the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_167.

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

10. **Configure initial synchronization.**

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

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

**Notice:**
- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
12. Close the Precheck dialog box after the following message is displayed: **The precheck is passed.** Then, the data synchronization task starts.

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

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

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

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

**Prerequisites**

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

**Limits**

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

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

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

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

**Supported synchronization topologies**

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

**Note:**
For more information about two-way synchronization, see [Configure two-way data synchronization between ApsaraDB RDS for MySQL instances](#).

**SQL operations that can be synchronized**

<table>
<thead>
<tr>
<th>Operation type</th>
<th>SQL statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DML</td>
<td>INSERT, UPDATE, DELETE, and REPLACE</td>
</tr>
</tbody>
</table>
### Operation type

<table>
<thead>
<tr>
<th>SQL statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDL</td>
</tr>
<tr>
<td>• ALTER TABLE and ALTER VIEW</td>
</tr>
<tr>
<td>• CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW</td>
</tr>
<tr>
<td>• DROP INDEX and DROP TABLE</td>
</tr>
<tr>
<td>• RENAME TABLE</td>
</tr>
<tr>
<td>• TRUNCATE TABLE</td>
</tr>
</tbody>
</table>

### Limits

- **Incompatibility of triggers**

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

  To avoid this situation, you must delete the trigger that is synchronized to the destination instance and select Table B as the object to be synchronized. For more information, see [Configure synchronization when triggers exist](#unique_142).

- **Limits on RENAME TABLE operations**

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

### Preparations

Before configuring the data synchronization task, you must create a database account and configure binary logging. For more information, see [#unique_142](#unique_142).

### Procedure

1. Purchase a data synchronization instance. For more information, see [#unique_166](#unique_166).

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

2. Log on to the **DTS console**.

3. In the left-side navigation pane, click **Data Synchronization**.

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

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

6. Configure the source and destination instances.

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Synchronization Task Name</td>
<td>DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Source Instance Details</td>
<td>Instance Type</td>
<td>Select <strong>User-Created Database in ECS Instance</strong>.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the source instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>ECS Instance ID</td>
<td>Select the ID of the ECS instance that is connected to the user-created MySQL database.</td>
</tr>
<tr>
<td></td>
<td>Database Type</td>
<td>The value of this parameter is set to <strong>MySQL</strong> and cannot be changed.</td>
</tr>
<tr>
<td></td>
<td>Port Number</td>
<td>Enter the service port number of the user-created MySQL database. The default port number is <strong>3306</strong>.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the account for the user-created MySQL database. The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the account of the user-created MySQL database.</td>
</tr>
<tr>
<td>Destination Instance Details</td>
<td>Instance Type</td>
<td>Select <strong>RDS Instance</strong>.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the destination instance. The region is the same as the region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>Instance ID</td>
<td>Select the ID of the destination RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account for the destination RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
</tbody>
</table>

**Note:**
If the database engine of the destination RDS instance is **MySQL 5.5** or **MySQL 5.6**, you do not need to configure the database account or database password.
## Section: Encryption

Select **Non-encrypted** or **SSL-encrypted**. If you want to select **SSL-encrypted**, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see [Configure SSL encryption for an RDS for MySQL instance](#).

### Note:
The **Encryption** parameter is available only in mainland China and Hong Kong(China).

---

7. In the lower-right corner of the page, click **Set Whitelist and Next**.
8. Configure the synchronization policy and objects.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing Mode</td>
<td></td>
</tr>
<tr>
<td>In Existed Target Table</td>
<td>• <strong>Pre-check and Intercept</strong>: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Ignore</strong>: skips the precheck for identical table names in the source and destination databases.</td>
</tr>
<tr>
<td>Note:</td>
<td>If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see #unique_167.</td>
</tr>
<tr>
<td>Warning:</td>
<td>If you select <strong>Ignore</strong>, data consistency is not guaranteed and your business may be exposed to potential risks.</td>
</tr>
<tr>
<td></td>
<td>- If the source and destination databases have the same schema, and the primary key of a record in the destination database is the same as that in the source database, the record remains unchanged during initial data synchronization. However, the record is overwritten during incremental data synchronization.</td>
</tr>
<tr>
<td></td>
<td>- If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Objects to be synchronized</td>
<td>Select objects from the <strong>Available</strong> section and click the <strong>»</strong> icon to move the objects to the <strong>Selected</strong> section. You can select tables and databases as the objects to be synchronized.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>• If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.</td>
</tr>
<tr>
<td></td>
<td>• After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination instance by using the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_167.</td>
</tr>
</tbody>
</table>

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

10. Configure initial synchronization.

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

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

**Note:**

- Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.
7.2.6 Synchronize data from an ApsaraDB RDS for MySQL instance to a user-created MySQL database connected over Express Connect, VPN Gateway, or Smart Access Gateway

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

Prerequisites

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

Note:

We recommend that you make sure the version of the source and destination MySQL databases is the same.

- The user-created MySQL database is connected to Alibaba Cloud VPC over Express Connect, VPN Gateway, or Smart Access Gateway. For more information, see #unique_147.

Note:

DTS is allowed to access the VPC to which the user-created MySQL database belongs. For more information, see #unique_150.

12. Close the Precheck dialog box after the following message is displayed: The precheck is passed.

13. Wait until the initial synchronization is complete and the synchronization task is in the Synchronizing state.

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

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

Limits

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

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

- If the source database does not have primary keys or UNIQUE constraints, and fields are not required to be unique, duplicate data may exist in the destination database.

Supported synchronization topologies

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

For more information about synchronization topologies, see #unique_165.

**SQL operations that can be synchronized**

<table>
<thead>
<tr>
<th>Operation type</th>
<th>SQL statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DML</td>
<td>INSERT, UPDATE, DELETE, and REPLACE</td>
</tr>
<tr>
<td>DDL</td>
<td>- ALTER TABLE and ALTER VIEW&lt;br&gt;- CREATE FUNCTION, CREATE INDEX, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW&lt;br&gt;- DROP INDEX and DROP TABLE&lt;br&gt;- RENAME TABLE&lt;br&gt;- TRUNCATE TABLE</td>
</tr>
</tbody>
</table>

**Limits**

- Incompatibility of triggers

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

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

- Limits on RENAME TABLE operations

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

1. Create a data synchronization task. For more information, see #unique_166.

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

2. Log on to the DTS console.

3. In the left-side navigation pane, click **Data Synchronization**.

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

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Synchronization Task Name</td>
<td>DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Source Instance Details</td>
<td>Instance Type</td>
<td>Select <strong>RDS Instance</strong>.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the source instance that you select when purchasing the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>Instance ID</td>
<td>Select the ID of the source RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account of the source RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
</tbody>
</table>

**Note:** If the database type of the source RDS instance is **MySQL 5.5** or **MySQL 5.6**, you do not need to configure the database account or database password.
ApsaraDB for RDS

RDS MySQL Database / 7 Data Synchronization

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

<table>
<thead>
<tr>
<th>Destination Instance Details</th>
<th>Instance Type</th>
<th>Select User-Created Database Connected over Express Connect, VPN Gateway, or Smart Access Gateway.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Region</td>
<td>instance region</td>
<td>The region of the destination instance that you select when purchasing the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td>Peer VPC</td>
<td>peer vpc</td>
<td>Select the ID of the VPC that is connected to the user-created MySQL database.</td>
</tr>
<tr>
<td>Database Type</td>
<td>database type</td>
<td>This parameter is set to MySQL and cannot be modified.</td>
</tr>
<tr>
<td>IP Address</td>
<td>ip address</td>
<td>Enter the IP address of the user-created MySQL database.</td>
</tr>
<tr>
<td>Port Number</td>
<td>port number</td>
<td>Enter the service port number of the user-created MySQL database. The default port number is 3306.</td>
</tr>
<tr>
<td>Database Account</td>
<td>database account account</td>
<td>Enter the account for the user-created MySQL database.</td>
</tr>
<tr>
<td>Note:</td>
<td>The account must have the REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects.</td>
<td></td>
</tr>
<tr>
<td>Database Password</td>
<td>database password</td>
<td>Enter the password for the database account.</td>
</tr>
</tbody>
</table>

7. In the lower-right corner of the page, click Set Whitelist and Next.
8. Configure the synchronization policy and objects.
### Parameter: In Existed Target Table

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pre-check and Intercept</strong>: checks whether the destination database contains tables that have the same names as tables in the source database. If the destination database does not contain tables that have the same names as tables in the source database, the pre-check is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started.</td>
</tr>
</tbody>
</table>

**Note:**

If tables in the destination database have the same names as tables in the source database, and cannot be deleted or renamed, you can use the object name mapping feature. For more information, see #unique_167.

- **Ignore**: skips the precheck for identical table names in the source and destination databases.

**Warning:**

If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks.

- If the source and destination databases have the same schema, and the primary key of a record in the destination database is the same as that in the source database, the record remains unchanged during initial data synchronization. However, the record is overwritten during incremental data synchronization.
- If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects to be synchronized</td>
<td>Select objects from the <strong>Available</strong> section and click the icon to move the objects to the <strong>Selected</strong> section. You can select tables and databases as the objects to be synchronized.</td>
</tr>
</tbody>
</table>

**Note:**
- If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database.
- After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the name of an object in the destination instance by using the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_167.

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

10. Configure initial synchronization.

   ![Initial Synchronization](image)

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

11. In the lower-right corner of the page, click **Precheck**.
Before you can start the data synchronization task, a precheck is performed. You can start the data synchronization task only after the task passes the precheck.

- If the task fails to pass the precheck, click the icon next to each failed item to view details. Troubleshoot the issues based on the causes and run the precheck again.

12. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed.**

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

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

### 7.3 Synchronize data from MySQL to other databases

#### 7.3.1 Synchronize data from an ApsaraDB RDS MySQL instance to a MaxCompute project

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

**Prerequisites**

- MaxCompute is activated. For more information, see [Activate MaxCompute](#).
- A project is created in MaxCompute. For more information, see [Create a project](#).

**Precautions**

- DTS uses read and write resources of the source and destination databases during initial full data synchronization. This may increase the database load. If the database performance is unfavorable, the specification is low, or the data volume is large,
ApsaraDB for RDS

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

- Only table-level data can be synchronized.
- We recommend that you do not use gh-ost or pt-online-schema-change to perform DDL operations on objects during data synchronization. Otherwise, data synchronization may fail.
- MaxCompute does not support the PRIMARY KEY constraint. If network errors occur, DTS may synchronize duplicate data records to MaxCompute.

**Supported source database types**

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

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

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

**Note:**

If your source database is a user-created MySQL database, you must prepare the environments that are required for the source database. For more information, see #unique_143.

**SQL operations that can be synchronized**

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

Issue: 20200702
Synchronization process

1. Initial schema synchronization
   DTS synchronizes the schemas of the required objects from the source database to MaxCompute. During initial schema synchronization, DTS adds the _base suffix to the end of the source table name. For example, if the name of the source table is customer, the name of the table in MaxCompute is customer_base.

2. Initial full data synchronization
   DTS synchronizes the historical data of the table from the source database to the destination table in MaxCompute. For example, the customer table in the source database is synchronized to the customer_base table in MaxCompute. The data is the basis for subsequent incremental synchronization.

   **Note:**
   The destination table that is suffixed with _base is known as a full baseline table.

3. Incremental data synchronization
   DTS creates an incremental data table in MaxCompute. The name of the incremental data table is suffixed with _log, such as customer_log. Then, DTS synchronizes incremental data that is generated in the source database to the incremental data table in real time.

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

Procedure

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

1. Purchase a data synchronization instance. For more information, see #unique_166.

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

2. Log on to the DTS console.

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

![Synchronization Tasks Page](image)

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

6. Configure the source and destination instances.

![Configure Source and Destination](image)

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Synchronization Task Name</td>
<td>DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Source Instance Details</td>
<td>Instance Type</td>
<td>Select <strong>RDS Instance</strong>.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>Instance ID</td>
<td>Select the ID of the source RDS instance.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account of the source RDS instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> If the database engine of the source RDS instance is <strong>MySQL 5.5</strong> or <strong>MySQL 5.6</strong>, you do not need to configure the <strong>database account</strong> or <strong>database password</strong>.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the source database account.</td>
</tr>
<tr>
<td></td>
<td>Encryption</td>
<td>Select <strong>Non-encrypted</strong> or <strong>SSL-encrypted</strong>. If you want to select <strong>SSL-encrypted</strong>, you must enable SSL encryption for the RDS instance before you configure the data synchronization task. For more information, see <a href="#">Configure SSL encryption for an RDS MySQL instance</a>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Notice:</strong> The <strong>Encryption</strong> parameter is available only for regions in mainland China and the Hong Kong (China) region.</td>
</tr>
<tr>
<td></td>
<td>Instance Type</td>
<td>The value of this parameter is set to <strong>MaxCompute</strong> and cannot be changed.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the destination instance. The region is the same as the destination region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>Project</td>
<td>Enter the name of the MaxCompute project. You can search for a project on the Workspaces page in the DataWorks console.</td>
</tr>
</tbody>
</table>

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

**Note:**
The CIDR blocks of DTS servers are automatically added to the whitelist of the source RDS instance and the MaxCompute project. This ensures that DTS servers can connect to the source and destination instances.
8. In the lower-right corner of the page, click **Next**. In this step, the permissions on the MaxCompute project are granted to the synchronization account.

To synchronize data to a MaxCompute instance, you must grant the following permissions of project detail to the synchronization account:

- CreateTable
- CreateInstance
- CreateResource
- CreateJob
- List

9. Configure the synchronization policy and objects.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition Definition of Incremental Data Table</td>
<td>Select the partition name based on your business requirements. For more information about partitions, see <strong>Partition</strong>.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Initial Synchronization</td>
<td>Initial synchronization includes initial schema synchronization and initial full data synchronization. Select both Initial Schema Synchronization and Initial Full Data Synchronization. In this case, DTS synchronizes the schemas and historical data of the required objects from the source database to the destination database before synchronizing incremental data.</td>
</tr>
</tbody>
</table>
| Processing Mode In Existed Target Table | • **Pre-check and Interception**: checks whether the destination database contains tables that have the same names as tables in the source database. If the source and destination databases do not contain identical table names, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started.  
  
  **Note:**
  You can change the names of the tables to be synchronized by using the object name mapping feature. You can use this feature if the source and destination databases contain identical table names and tables in the destination database cannot be deleted or renamed. For more information, see #unique_167.
  
> • **Ignore**: skips the precheck for identical table names in the source and destination databases.  
  
> **Warning:**
> If you select Ignore, data consistency is not guaranteed and your business may be exposed to potential risks.

- DTS does not synchronize data records that have the same primary keys as data records in the destination database during initial data synchronization. This occurs if the source and destination databases have the same schema. However, DTS synchronizes these data records during incremental data synchronization.
- If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Note:</th>
<th></th>
</tr>
</thead>
</table>
| Objects to be synchronized      | Select tables from the Available section and click the icon to move the tables to the Selected section.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | • You can select tables from multiple databases as the objects to be synchronized.  
• After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the names of the objects that are synchronized to the destination database by using the object name mapping feature. For more information about how to use this feature, see #unique_167. |  |

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

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

11. Close the Precheck dialog box after the following message is displayed: The precheck is passed.

12. Wait until the initial synchronization is complete and the data synchronization task is in the Synchronizing state.

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

---

### Issue: 20200702

---
### Schema of an incremental data table

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>record_id</td>
<td>The ID of the incremental log entry.</td>
</tr>
<tr>
<td>operation_flag</td>
<td>The operation type. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• I: an INSERT operation.</td>
</tr>
<tr>
<td></td>
<td>• D: a DELETE operation.</td>
</tr>
<tr>
<td></td>
<td>• U: an UPDATE operation.</td>
</tr>
<tr>
<td>utc_timestamp</td>
<td>The operation timestamp. It is also the timestamp of the binary log file.</td>
</tr>
<tr>
<td></td>
<td>The timestamp is in the UTC format.</td>
</tr>
<tr>
<td>before_flag</td>
<td>Indicates whether the column values are pre-update values. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• Y and N.</td>
</tr>
<tr>
<td>after_flag</td>
<td>Indicates whether the column values are post-update values. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• Y and N.</td>
</tr>
</tbody>
</table>

**Note:**

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

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

- **INSERT**

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

- **UPDATE**

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

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

- **DELETE**

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

**Merge a full baseline table and incremental data table**

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

This section describes how to merge data for the customer table. The following figure shows the schema of the customer table.
1. Create a table in MaxCompute based on the schema of the source table. The table is used to store the merged data.

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

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

Note:
- You can use the ad-hoc query feature to run SQL statements. For more information, see (Optional) Use an ad-hoc query to run SQL statements.
- For more information about the data types that are supported by MaxCompute, see Data types.

2. Run the following SQL statements in MaxCompute to merge the full baseline table and incremental data table and obtain full data at a specific time point:

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

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

Run the following SQL statements to obtain full data of the customer table at the 1565944878 time point:

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

3. Query the merged data from the customer_1565944878 table.
7.3.2 Synchronize data from an ApsaraDB RDS for MySQL instance to an AnalyticDB for MySQL cluster

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

Prerequisites

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

Precautions

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

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

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

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

Note:
If your source database is a user-created MySQL database, you must prepare the environments that are required for the source database. For more information, see #unique_143.

SQL operations that can be synchronized

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

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

Permissions required for database accounts

<table>
<thead>
<tr>
<th>Database</th>
<th>Required permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApsaraDB RDS for MySQL</td>
<td>The REPLICATION SLAVE permission, the REPLICATION CLIENT permission, the SHOW VIEW permission, and the permission to perform SELECT operations on the required objects</td>
</tr>
<tr>
<td>AnalyticDB for MySQL</td>
<td>The read/write permissions on the objects to be synchronized</td>
</tr>
</tbody>
</table>
Data type mappings

The data types of ApsaraDB RDS for MySQL and AnalyticDB for MySQL do not have one-to-one correspondence. During initial schema synchronization, DTS converts the data types of the source database into those of the destination database. For more information, see #unique_173.

Procedure

1. #unique_166/unique_166_Connect_42_section_39h_fto_gdl.

   ![Note]

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

2. Log on to the DTS console.

3. In the left-side navigation pane, click Data Synchronization.

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

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Synchronization Task Name</td>
<td>DTS automatically generates a task name. We recommend that you specify an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Source Instance Details</td>
<td>Instance Type</td>
<td>Select RDS Instance.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the source instance. The region is the same as the source region that you selected when you purchased the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>Instance ID</td>
<td>Select the ID of the source RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account of the source RDS instance. For more information about the permissions that are required for the account, see Permissions required for database accounts.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the source database account.</td>
</tr>
</tbody>
</table>

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

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

![Configure Source and Destination Instances](image)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Synchronization</td>
<td>You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination cluster. The schemas and data are the basis for subsequent incremental synchronization.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Processing Mode In Existed Target Table | • **Pre-check and Intercept**: checks whether the destination database contains tables that have the same names as tables in the source database. If the source and destination databases do not contain identical table names, the precheck is passed. Otherwise, an error is returned during precheck and the data synchronization task cannot be started. 

**Note:**
You can change the names of the tables to be synchronized by using the object name mapping feature. You can use this feature if the source and destination databases contain identical table names and tables in the destination database cannot be deleted or renamed. For more information, see #unique_167.

• **Ignore**: skips the precheck for identical table names in the source and destination databases.

**Warning:**
If you select **Ignore**, data consistency is not guaranteed and your business may be exposed to potential risks.
- If the source and destination databases have the same schema, DTS does not synchronize data records that have the same primary keys as data records in the destination database.
- If the source and destination databases have different schemas, initial data synchronization may fail. In this case, only some columns are synchronized or the data synchronization task fails. |
| Merge Multi Tables | • If you select **Yes**, DTS adds the `__dts_data_source` column to each table to record data sources. In this case, DDL operations cannot be synchronized.

• **No** is selected by default. In this case, DDL operations can be synchronized.

**Note:**
You can merge the data source columns based on tasks rather than tables. To merge only the data source columns of some tables, you can create two data synchronization tasks. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization Type</td>
<td>Select the types of operations that you want to synchronize based on your business requirements. All operation types are selected by default. Note: Only INSERT, UPDATE, DELETE, and ADD COLUMN operations can be synchronized.</td>
</tr>
<tr>
<td>Objects to be synchronized</td>
<td>Select objects from the Available section and click the icon to move the objects to the Selected section. You can select tables and databases as the objects to be synchronized. Note: • If you select a database as the object to be synchronized, all schema changes in the database are synchronized to the destination database. • If you select a table as the object to be synchronized, only ADD COLUMN operations on the table are synchronized to the destination database. • After an object is synchronized to the destination database, the name of the object remains unchanged. You can change the names of the objects that are synchronized to the destination cluster by using the object name mapping feature. For more information about how to use this feature, see #unique_167.</td>
</tr>
</tbody>
</table>

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

10. Specify a type for the tables to be synchronized to the destination database.
After you select **Initial Schema Synchronization**, you must specify the **type**, **primary key column**, and **partition key column** for the tables to be synchronized to AnalyticDB for MySQL. For more information, see [CREATE TABLE](#).

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

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

12. Close the **Precheck** dialog box after the following message is displayed: **The precheck is passed**.

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

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

Troubleshoot the synchronization failure that occurs due to field type changes

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

1. In the destination AnalyticDB for MySQL cluster, create a table named customer_new with the same schema as the customer table.

2. Run the `INSERT INTO SELECT` command to copy the data of the customer table and insert the data into the customer_new table. This ensures that the data of the two tables is consistent.

3. Rename or delete the customer table. Then, change the name of the customer_new table to customer.
4. Restart the data synchronization task in the DTS console.

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

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

Prerequisites

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

Notes

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

Limits

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

Supported SQL operations

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

Note:
The CREATE TABLE operation is not supported. To synchronize data from a new table, you must add the table to the selected objects. For more information, see #unique_175.

Supported synchronization topologies

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

Term mappings

<table>
<thead>
<tr>
<th>Term in MySQL</th>
<th>Term in AnalyticDB for PostgreSQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Schema</td>
</tr>
<tr>
<td>Table</td>
<td>Table</td>
</tr>
</tbody>
</table>

Procedure

1. Purchase a data synchronization instance. For more information, see #unique_166.

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

2. Log on to the DTS console.

3. In the left-side navigation pane, click Data Synchronization.

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

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Synchronization Task Name</td>
<td>DTS automatically generates a task name. We recommend that you use an informative name for easy identification. You do not need to use a unique task name.</td>
</tr>
<tr>
<td>Source Instance Details</td>
<td>Instance Type</td>
<td>Select RDS Instance.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the source instance. The value is the same as that you selected when purchasing the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>Instance ID</td>
<td>Select the ID of the source RDS instance.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account for the source ApsaraDB RDS for MySQL instance.</td>
</tr>
<tr>
<td></td>
<td>Note:</td>
<td>If the database type of the source RDS instance is MySQL 5.5 or MySQL 5.6, you do not need to configure the database account and database password.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
<tr>
<td></td>
<td>Encryption</td>
<td>Select <strong>Non-encrypted</strong> or <strong>SSL-encrypted</strong>. If you want to select <strong>SSL-encrypted</strong>, you must enable SSL encryption for the RDS instance before configuring the data synchronization task. For more information, see <a href="#">Configure SSL encryption for an RDS for MySQL instance</a>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The <strong>Encryption</strong> parameter is available only in mainland China and Hong Kong(China).</td>
</tr>
<tr>
<td><strong>Destination Instance Details</strong></td>
<td>Instance Type</td>
<td>The value of this parameter is set to <strong>AnalyticDB for PostgreSQL</strong> and cannot be changed.</td>
</tr>
<tr>
<td></td>
<td>Instance Region</td>
<td>The region of the destination instance. The value is the same as that you selected when purchasing the data synchronization instance. You cannot change the value of this parameter.</td>
</tr>
<tr>
<td></td>
<td>Instance ID</td>
<td>Select the ID of the destination AnalyticDB for PostgreSQL instance.</td>
</tr>
<tr>
<td></td>
<td>Database Name</td>
<td>The name of the destination database.</td>
</tr>
<tr>
<td></td>
<td>Database Account</td>
<td>Enter the database account for the destination AnalyticDB for PostgreSQL instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The database account must have the SELECT, INSERT, UPDATE, DELETE, COPY, TRUNCATE, and ALTER TABLE permissions.</td>
</tr>
<tr>
<td></td>
<td>Database Password</td>
<td>Enter the password for the database account.</td>
</tr>
</tbody>
</table>

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

**Note:**
The CIDR blocks of DTS servers are automatically added to the whitelists of the source and destination instances. This ensures that DTS servers can connect to the source and destination instances.
8. Configure the synchronization policy and objects.

<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization policy</td>
<td>Initial Synchronization</td>
<td>You must select both Initial Schema Synchronization and Initial Full Data Synchronization in most cases. After the precheck, DTS synchronizes the schemas and data of the required objects from the source instance to the destination instance. The schemas and data are the basis for subsequent incremental synchronization.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Processing Mode In</td>
<td>In Existed Target Table</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Clear Target Table</strong></td>
<td>Skips the <strong>Schema Name Conflict</strong> item during the precheck. Clears the data in the destination table before initial full data synchronization. If you want to synchronize your business data after testing the data synchronization task, you can select this mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Ignore</strong></td>
<td>Skips the <strong>Schema Name Conflict</strong> item during the precheck. Adds new data to the existing data during initial full data synchronization. You can select this mode if you want to synchronize data from multiple tables to one table.</td>
</tr>
<tr>
<td>Synchronization Type</td>
<td><strong>Select the types of operations that you want to synchronize based on your business requirements.</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Insert</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Update</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Delete</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Alter Table</strong></td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Objects to be synchronized</td>
<td>N/A</td>
<td>Select tables from the Available section and click the right arrow (&gt;) icon to add the tables to the Selected section.</td>
</tr>
</tbody>
</table>

**Note:**
- You can select only tables as the objects to be synchronized.
- You can change the names of columns in the destination database by using the object name mapping feature provided by DTS. For more information about how to use this feature, see #unique_167.

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

![Diagram of configuration steps](image)

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

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

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

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

On the Synchronization Tasks page, view the status of the data synchronization task.
8 Manage pending events

This topic describes how to manage pending events. When an ApsaraDB for RDS event is pending for processing, you will be notified of handling the event in the console.

You can receive notifications for events such as instance migration and version upgrade by SMS messages, phone calls, emails, internal messages, and console notifications. You can view event types, regions, procedures, precautions, and related instances as well as change the scheduled switchover time.

Prerequisites

A pending O&M event exists.

Note:
If there are unprocessed O&M events, you can see notification badges on the Pending Events button in the upper-right corner of the page.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. In the left-side navigation pane or the upper-right corner, click Pending Events.

Note:
For an O&M event for which you must reserve the switch time, a dialog box appears, prompting you to complete the reservation as soon as possible.

3. On the Pending Events page, select the type and region of the event that you want to handle.

Note:
The notification, such as the event process and precautions, varies depending on Event Type.
4. View event details in the instance list. To change the **Scheduled Disconnection Time**, select an event, and then click **Specify Disconnection Time**. In the dialog box that appears, set the switch time, and then click **OK**.

**Note:**
- The displayed information varies depending on the event type.
- The **Scheduled Disconnection Time** cannot be later than the time specified in the **Set Before** column.

### Event types

<table>
<thead>
<tr>
<th>Event type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>rds_apsaradb_db_backup</code></td>
<td>Enable individual database or table restoration.</td>
</tr>
<tr>
<td><code>rds_apsaradb_ha</code></td>
<td>Switch services from a primary instance to a secondary instance.</td>
</tr>
<tr>
<td><code>rds_apsaradb_network_mode_upgrade</code></td>
<td>Upgrade the instance network mode (dedicated proxy supported).</td>
</tr>
<tr>
<td><code>rds_apsaradb_network_upgrade</code></td>
<td>Upgrade the network bandwidth.</td>
</tr>
<tr>
<td><code>rds_apsaradb_release_instances</code></td>
<td>Delete an expired instance.</td>
</tr>
<tr>
<td><code>rds_apsaradb_ssl_update</code></td>
<td>Upgrade the SSL certificate.</td>
</tr>
<tr>
<td><code>rds_apsaradb_switch_proxy_tenant</code></td>
<td>Upgrade a shared proxy to a dedicated proxy.</td>
</tr>
<tr>
<td><code>rds_apsaradb_transfer</code></td>
<td>Migrate an instance across zones.</td>
</tr>
<tr>
<td><code>rds_apsaradb_upgrade</code></td>
<td>Upgrade the minor version.</td>
</tr>
<tr>
<td><code>rds_apsaradb_maxscale</code></td>
<td>Upgrade the minor version of a dedicated proxy.</td>
</tr>
</tbody>
</table>
9.1 Create an ApsaraDB RDS for MySQL instance

This topic describes how to create an ApsaraDB RDS for MySQL instance in the ApsaraDB for RDS console. You can also call an API operation to create an ApsaraDB RDS for MySQL instance.

For information about how to create an instance in other database engines, see the following topics:

- #unique_74
- #unique_75
- #unique_76
- #unique_77

A new ApsaraDB for RDS console is available for instance creation. You can choose the new console or switch back to the original console.

Billing

For more information, see Pricing, billing items, and billing methods.

Prerequisites

You have an Alibaba Cloud account. For more information, see Sign up with Alibaba Cloud.

Precautions

- Subscription instances cannot be converted into pay-as-you-go instances.
- Pay-as-you-go instances can be converted into subscription instances. For more information, see Switch the billing method from pay-as-you-go to subscription.
- You can create up to 30 pay-as-you-go instances in your Alibaba Cloud account. To increase this quota, submit a ticket.

Create an RDS instance in the new ApsaraDB for RDS console

1. Log on to the new ApsaraDB for RDS console.

Note:
You can click **Back to Old Version** to switch back to the original ApsaraDB for RDS console.

2. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Billing Method  | **Subscription**: You must pay the subscription fee when you create an instance. We recommend that you choose subscription billing for long-term use because it is more cost-effective than pay-as-you-go billing. You receive larger discounts for longer subscription periods.  
**Pay-As-You-Go**: A pay-as-you-go instance is charged per hour based on your actual resource usage. We recommend that you choose pay-as-you-go billing for short-term use. You can release your pay-as-you-go instance to reduce costs when you no longer need it. |
| Region          | The region where the RDS instance resides.  
- You cannot change the region after you confirm your order.  
- To maximize access speed, we recommend that you select a region in close proximity to the geographic location where your users reside.  
- Make sure that the RDS instance is created in the same region as the ECS instance to which you want to connect. If the RDS and ECS instances reside in different regions, they cannot communicate over an internal network and therefore cannot deliver optimal performance. |
| Database Engine | The database engine and version that the RDS instance runs. Select the **MySQL** database engine. Supported MySQL versions are 5.5, 5.6, 5.7, and 8.0.  
**Note**: The database engines and versions available vary based on the region you select. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Edition                   | • **Basic**: Your database system only consists of one instance. Computing is separated from storage to increase cost-effectiveness.  
• **High-availability**: Your database system works in the classic high-availability architecture, which consists of one primary instance and one secondary instance.  
• **Enterprise Edition**: Your database system consists of one primary instance and two secondary instances, which are located in three different zones within the same region to provide finance-level reliability. |
|                           | **Note:** The RDS editions available vary based on the region and database engine version you select. For more information, see [ApsaraDB for RDS edition overview](https://example.com).                                                                                                            |
| Storage Type              | • **Local SSD**: A local SSD is housed on the same server as the database engine. If you use local SSDs, computing is closer to data to reduce I/O latency.  
• **Enhanced SSD**: An enhanced SSD is an ultra-high performance disk provided by Alibaba Cloud based on the new generation of distributed block storage architecture. It integrates 25 Gigabit Ethernet and remote direct memory access (RDMA) technologies to reduce latency and deliver up to 1 million random input/output operations per second (IOPS).  
• **Standard SSD**: A standard SSD is an elastic block storage device designed based on the distributed storage architecture. If you use standard SSDs, computing is separated from storage. For more information, see [Storage types](https://example.com). |
| Zone of Primary Node      | The zone of primary node where the RDS instance resides. Each zone is an independent physical location within a region. There is no substantive difference between zones in the same region. If your database system spans multiple zones, it provides zone-level disaster recovery.  
You only need to select a primary zone. The system automatically assigns a secondary zone to the RDS instance. |
<p>| Deployment Method         | You can set the instance as multi-zone or single-zone deployment.                                                                                                                                                                                                                                                                         |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Instance Type   | • **Entry-level**: belongs to the general-purpose instance family. A general-purpose instance occupies the exclusive memory and I/O resources allocated to it, but shares CPU and storage resources with the other general-purpose instances that are deployed on the same server.  
• **Enterprise-level**: belongs to the dedicated instance family. A dedicated instance occupies the exclusive CPU, memory, storage, and I/O resources allocated to it. The top configuration of the dedicated instance family is the dedicated host. A dedicated host instance occupies all the CPU, memory, storage, and I/O resources on the server where it is housed. |
| Capacity        | The storage capacity used to store data files, system files, binary log files, and transaction files. The storage capacity increases in increments of 5 GB. |

**Note:**
Each instance type supports a specific number of CPU cores, memory capacity, maximum number of connections, and maximum IOPS. For more information, see #unique_82.

**Note:**
The dedicated instance family supports exclusive allocations of resources. Therefore, the storage capacity of each instance type with local SSDs in this family is fixed. For more information, see #unique_82.

3. Click **Next: Instance Configuration**.
4. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Network Type**   | - **Classic Network**: a traditional type of network.  
- **VPC**: A virtual private cloud (VPC) is an isolated virtual network with higher security and better performance than a classic network. If you select the VPC network type, you must also specify **VPC** and **VSwitch of Primary Node**.  

**Note:**  
The primary RDS instance must have the same network type as the ECS instance to which you want to connect. If both their network types are VPC, you must also make sure that they reside in the same VPC. Otherwise, they cannot communicate over an internal network. |
| **Storage Engine** | The default storage engine of the instance. This parameter is only available to instances that run MySQL 8.0 in the High-availability Edition equipped with local SSDs. For information about the X-Engine developed by Alibaba Cloud.  

**Note:**  
X-Engine is compatible with InnoDB but performs better than InnoDB. We recommend that you select X-Engine as the default storage engine. |
| **Parameter Template** | The parameter template of the instance. You can select a system parameter template or create a custom parameter template. For more information, see Use a parameter template to manage parameters.  

**Note:**  
This parameter is only available to instances in High-availability Edition or Basic Edition. |
| **Time Zone**      | The time zone of the instance.  

**Note:**  
This parameter is only available to instances in High-availability Edition or Basic Edition. |
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table Name Case Sensitivity</strong></td>
<td>Specifies whether table names on the instance are case-sensitive. If table names in your on-premises database are case-sensitive, we recommend that you select <strong>Case-sensitive</strong>, so you can migrate data between the instance and your on-premises database with ease.</td>
</tr>
</tbody>
</table>

**Note:**
This parameter is only available to instances in High-availability Edition or Basic Edition.

5. Click **Next: Confirm Order**.

6. Confirm the settings in the **Parameters** section, specify **Purchase Plan** and **Duration**, read and select Terms of Service, and click **Pay Now**. You only need to specify Duration when you create a subscription instance.

---

### Create an RDS instance in the original ApsaraDB for RDS console

1. Log on to the [original ApsaraDB for RDS console](https://example.com).

2. Select a billing method.
   - **Pay-As-You-Go**: A pay-as-you-go instance is charged per hour based on your actual resource usage. We recommend that you choose pay-as-you-go billing for short-term use. You can release your pay-as-you-go instance to reduce costs when you no longer need it.
   - **Subscription**: You must pay the subscription fee when you create an instance. We recommend that you choose subscription billing for long-term use because it is more cost-effective than pay-as-you-go billing. You receive larger discounts for longer subscription periods.
3. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region</td>
<td>The region where the RDS instance resides. You cannot change the region after you confirm your order.</td>
</tr>
<tr>
<td></td>
<td>• To maximize access speed, we recommend that you select a region in close proximity to the geographic location where your users reside.</td>
</tr>
<tr>
<td></td>
<td>• Make sure that the RDS instance is created in the same region as the ECS instance to which you want to connect. If the RDS and ECS instances reside in different regions, they cannot communicate over an internal network and therefore cannot deliver optimal performance.</td>
</tr>
<tr>
<td>Resource Group</td>
<td>The resource group to which the RDS instance belongs.</td>
</tr>
<tr>
<td>Database Engine</td>
<td>The database engine that the RDS instance runs. Select <strong>MySQL</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>The database engines available vary based on the region you select.</td>
</tr>
<tr>
<td>Version</td>
<td>The version of the database engine that the RDS instance runs. Supported MySQL versions are 5.5, 5.6, 5.7, and 8.0.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>The database engine versions available vary based on the region you select.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Edition</strong></td>
<td>• <strong>Basic</strong>: Your database system only consists of one instance. Computing is separated from storage to increase cost-effectiveness.</td>
</tr>
<tr>
<td></td>
<td>• <strong>High-availability</strong>: Your database system works in the classic high-availability architecture, which consists of one primary instance and one secondary instance.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Enterprise Edition</strong>: Your database system consists of one primary instance and two secondary instances, which are located in three different zones within the same region to provide finance-level reliability. The Enterprise Edition is only available in four regions: China (Hangzhou), China (Shanghai), China (Shenzhen), and China (Beijing).</td>
</tr>
<tr>
<td><strong>Storage Type</strong></td>
<td>• <strong>Local SSD</strong>: A local SSD is a storage device in the server node where the database engine resides. If you use local SSDs, computing is closer to data to reduce I/O latency.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Standard SSD</strong>: A standard SSD is an elastic block storage device designed based on the distributed storage architecture. If you use standard SSDs, computing is separated from storage.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Enhanced SSD</strong>: An enhanced SSD is an ultra-high performance disk provided by Alibaba Cloud based on the new generation of distributed block storage architecture. It integrates 25 Gigabit Ethernet and RDMA technologies to reduce latency and deliver up to 1 million random IOPS. For more information, see Storage types.</td>
</tr>
<tr>
<td><strong>Zone</strong></td>
<td>The zone where the RDS instance resides. Each zone is an independent physical location within a region. There is no substantive difference between zones in the same region. You can deploy your primary and secondary instances in the same zone or in different zones.</td>
</tr>
<tr>
<td></td>
<td>If your database system spans multiple zones, it provides zone-level disaster recovery.</td>
</tr>
</tbody>
</table>
### Parameter | Description
--- | ---
**Network Type** | • **Classic Network**: a traditional type of network.  
• **VPC**: This is the recommended network type. A VPC is an isolated virtual network with higher security and better performance than a classic network.  

**Note:**  
The RDS instance must have the same network type as the ECS instance to which you want to connect. If the RDS and ECS instances have different network types, they cannot communicate over an internal network.

**CPU and Memory** | Each instance type supports a specific number of CPU cores, memory capacity, maximum number of connections, and maximum IOPS. For more information, see #unique_82.

ApsaraDB for RDS provides the following instance families:

• **General-purpose instance**: A general-purpose instance occupies the exclusive memory and I/O resources allocated to it, but shares CPU and storage resources with the other general-purpose instances that are deployed on the same server.  
• **Dedicated instance**: A dedicated instance occupies the exclusive CPU, memory, storage, and I/O resources allocated to it.  
• **Dedicated host**: This is the top configuration of the dedicated instance family. A dedicated host instance occupies all the CPU, memory, storage, and I/O resources on the server where it is housed.

For example, "8 Cores, 32 GB" is a general-purpose instance type, "8 Cores, 32 GB (Dedicated Instance)" is a dedicated instance type, and "30 Cores, 220 GB (Dedicated Host)" is a dedicated host instance type.

**Capacity** | The storage capacity used to store data files, system files, binary log files, and transaction files.

4. **Specify Duration and Quantity, and click Buy Now.** You only need to specify Duration when you create a subscription instance.

**Note:**  
• When you create a subscription instance, you can select **Auto-renewal**, so the system automatically renews your subscription based on the duration you specify.
For example, if you create a three-month subscription instance with **Auto-renewal** selected, you are charged a three-month subscription fee for each automatic renewal.

- When you create a subscription instance, you can click **Add to Cart** to add it to the shopping cart. Later, you can click **Cart** to pay for the subscription instance.

5. On the **Order Confirmation** page, confirm the instance configuration, read and select Product Terms of Service, and click Pay Now.

**What to do next**

In the top navigation bar, select the region where the RDS instance you just created resides.

After the RDS instance is created, you must **configure whitelists** and **create accounts** for it. (If you want to connect to the RDS instance over the Internet, you must also **apply for a public endpoint**.) After you complete these operations, you can **connect to the RDS instance**.
FAQ

• After I purchase an RDS instance, the ApsaraDB for RDS console does not respond nor can I find the instance I just created. Why?

There are two possible reasons:

- The RDS instance you created does not reside in the region you selected.

   In the top navigation bar, select the region where the RDS instance resides. Then you can find the RDS instance you just created.

- The zone you selected cannot provide sufficient resources.

   Resources in zones are dynamically allocated. After you confirm the purchase order, the zone you selected may fail to provide sufficient resources. As a result, the instance creation fails. We recommend that you select another zone and try again. You can visit the order list to check that the fees are refunded to you.

• How do I authorize a RAM user to manage an RDS instance?

   For more information, see Use RAM to manage ApsaraDB for RDS permissions.

Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_83</td>
<td>Creates an ApsaraDB for RDS Instance.</td>
</tr>
</tbody>
</table>

9.2 Restart an RDS MySQL instance

This topic describes how to restart an RDS MySQL instance in the RDS console if the number of connections exceeds its upper limit or any performance issue occurs for the instance.

Impact

Restarting an RDS instance may interrupt its connections and impact your services. Exercise caution when performing this action.
Procedure

1. Log on to the RDS console.
2. Select the target region.
3. Find the target RDS instance. Then, click the instance ID or in the Actions column click Manage.
4. In the upper-right corner of the Basic Information page, click Restart Instance.
5. In the displayed dialog box, click Confirm.

APIs

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_178</td>
<td>Used to restart an RDS instance.</td>
</tr>
</tbody>
</table>

9.3 Renew instance

9.3.1 Manually renew an RDS MySQL instance

This topic describes how to manually renew an RDS MySQL instance that is charged by using subscription billing. If a subscription RDS instance expires and is not renewed in time, services will be stopped and data may be permanently deleted.

For more information about the impacts, see Expiration and overdue policy.
Note:
A pay-as-you-go instance does not have an expiration date and no renewal is required.

You can manually renew a subscription-based instance before it expires or within 15 days after it expires.

Method 1: Renew an RDS instance in the RDS console

1. Log on to the RDS console.
2. Select the target region.
3. Find the target RDS instance and in the Actions column click Renew.
4. On the Renew Subscription page, select a duration. The longer the duration, the bigger discount you have.
5. Read and confirm you agree to Terms of Service, Service Level Agreement, and Terms of Use by selecting the checkbox, confirm the order details, and click Pay Now.

Renew an RDS instance in the Renew console

1. Log on to the RDS console.
2. In the upper-right corner of the page, choose **Billing Management > Renew**.

3. In the left-side navigation pane, click **ApsaraDB for RDS**.

4. On the **Renew** tab, find the target RDS instance and in the **Actions** column click **Renew**.

   ![Renew tab](image)

   **Note:**
   - If the target RDS instance is on the **Nonrenewal** tab, you can click **Enable Manual Renew** in the **Actions** column to restore the instance to manual renewal.
   - If the target RDS instance is on the **Auto** tab, you can click **Enable Manual Renew** in the **Actions** column to restore the instance to manual renewal.

5. Select a duration, read and confirm you agree to **Terms of Service, Service Level Agreement, and Terms of Use** by selecting the checkbox, confirm the order details, and click **Pay Now**.

**Auto-renewal**

Enabling auto-renewal guarantees that your business runs smoothly without the need of manual renewal when your instance expires. For more information, see [Automatically renew an RDS MySQL instance](#).

### 9.3.2 Automatically renew an RDS MySQL instance

This topic describes how to automatically renew an RDS MySQL instance.

Each subscription-based instance has an expiration date. If an instance is not renewed in time when the instance expires, a service interruption or even data loss may occur. For more information about the impacts, see [Expiration and overdue policy](#). Enabling auto-renewal guarantees that your business runs smoothly without the need of manual renewal when your instance expires.
Note:
A pay-as-you-go-based instance does not have an expiration date and no renewal is required.

Precautions

- If you have enabled automatic renewal for your subscription-based instance, a payment will be deducted three days before the expiration date. You can pay the fees by credit cards or coupons. Make sure that your credit card has sufficient balance.
- If you manually renew an instance before the automatic deduction date, the system will automatically renew the instance before the next expiration date.
- The automatic renewal function takes effect the next day after you enable it. If your instance expires the next day, renew it manually to prevent service interruption. For more information, see Manually renew an RDS MySQL instance.

Enable automatic renewal when you purchase an RDS instance

When you purchase a subscription-based instance, you can select Auto Renewal on the purchase page.

Enable automatic renewal after you purchase an RDS instance

Note:
After you enable automatic renewal, the system automatically renews your instance based on the selected renewal duration. For example, if you select a three-month renewal duration, the fees are automatically paid every three months for each renewal.

1. Log on to the RDS console.
2. In the upper-right corner, choose Billing Management > Renew.
3. In the left-side navigation pane, click ApsaraDB for RDS.

4. On the Manually Renew or Auto-Renew tab, find the target RDS instance. You can enable automatic renewal for one or more RDS instances at a time.

   - Follow these steps to enable automatic renewal for one RDS instance:
     a. Find the target RDS instance and in the Actions column click Enable Auto-Renew.
     b. In the displayed dialog box, set Auto-Renew Cycle and click Enable Auto-Renew.

   - Follow these steps to enable auto-renewal for more than one RDS instance:
     Select the target RDS instances, and click Enable Auto-Renew below the instance list.
     • In the displayed dialog box, set Auto-Renew Cycle and click Enable Auto-Renew.

Change the auto-renew cycle of an RDS instance

1. Log on to the RDS console.

2. In the upper-right corner, choose Billing Management > Renew.

3. In the left-side navigation pane, click ApsaraDB for RDS.

4. On the Auto tab, find the target RDS instance and in the Actions column click Edit Auto Renewal.

5. Select a renewal duration, and click OK.
Disable automatic renewal for an RDS instance

1. Log on to the RDS console.
2. In the upper-right corner, choose **Billing Management > Renew**.
3. In the left-side navigation pane, click **ApsaraDB for RDS**.
4. On the **Auto** tab, find the target RDS instance and in the **Actions** column click **Enable Manual Renewal**.
5. Click **OK**.

### APIs

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_83</td>
<td>Used to create an RDS instance.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>Automatic renewal is enabled when you create the instance.</td>
</tr>
<tr>
<td>#unique_183</td>
<td>Used to renew a subscription-based RDS instance.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>Automatic renewal is enabled after you create the instance.</td>
</tr>
</tbody>
</table>
9.4 Release an RDS MySQL instance

This topic describes how to release an RDS MySQL instance, which can use the pay-as-you-go or subscription billing method.

Note:
After an RDS instance is released, its data is deleted immediately. We recommend that you back up the instance data before you release the instance.

Release a pay-as-you-go RDS instance

Precautions

If the RDS instance you want to release is the last read-only instance for a master RDS instance, you must first disable the read/write splitting function. For more information, see Disable read/write splitting for an RDS MySQL instance.

Procedure

1. Log on to the RDS console.
2. Select the target region.
3. Use one of the following two methods to open the **Release Instance** dialog box:

- **Method 1:**
  
  Find the target RDS instance and in the **Actions** column choose **More > Release Instance**.

- **Method 2:**
  
  a. Find the target RDS instance and click the instance ID.
  
  b. On the **Basic Information** page, find the **Status** section and click **Release Instance**.

4. In the **Release Instance** dialog box, click **Confirm**.

**Release a subscription RDS instance**

You can open a ticket to apply for releasing a subscription RDS instance.

**FAQ**

Will my RDS services be affected if I release a read-only instance?

Yes, we recommend that you set the **read weight** of the read-only instance to 0 before you release a read-only instance.

**Note:**

After you release a read-only instance, the cached connection still points to this read-only instance. To enable the system to distribute read requests to the other read-only instances, you must reestablish a connection.
9.5 Manage ApsaraDB RDS for MySQL instances that are in the recycle bin

Expired or overdue ApsaraDB for RDS instances are locked in the recycle bin. You can unlock, rebuild, or destroy them in the recycle bin.

Unlock an overdue pay-as-you-go instance

If a pay-as-you-go instance is locked due to overdue payments, check the billing method of your Alibaba Cloud account.

Unlock an expired subscription instance

If a subscription instance is locked because it is expired, you can go to the recycle bin to renew it.

1. Log on to the recycle bin.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click Unlock to renew it.

The target RDS instance is unlocked immediately after the renewal.

Rebuild a subscription instance

After a subscription instance expires, it is retained for a specific period of time. After the period of time you specify elapses, the instance is released. The data backup files of the

---

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_186</td>
<td>Used to release a pay-as-you-go RDS instance. (A subscription RDS instance cannot be released by calling an API action.)</td>
</tr>
</tbody>
</table>

---
instance are retained for eight days. During the eight-day retention period, you can use the data backup files to rebuild a new instance. For more information, see #unique_187.

1. Log on to the recycle bin.

2. In the top navigation bar, select the region where the target RDS instance resides.

3. Find the target RDS instance and click **Recreate Instance**.

By default, the target RDS instance is rebuilt with the same specifications and in the same zone. You also have the option to rebuild the target RDS instance with different specifications and in a different zone.

**Destroy an instance**

If an instance is locked due to expiration or overdue payments, you can destroy it in the recycle bin.

⚠️ **Warning:**

Destroying an instance will destroy all backups at the same time, including regular data backup, log backup, and permanent backup. In addition to cross-region backup, please be careful.

**Procedure**

1. Log on to the recycle bin.

2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click Destroy.

References

#unique_187
10 Instance Change

10.1 Change specifications

This topic describes how to change the specifications of an ApsaraDB RDS for MySQL instance, including the edition, instance type, and storage capacity.

For more information about how to change specifications in other database engines, see the following topics:

- Change the specifications of an ApsaraDB RDS for SQL Server instance
- Change the specifications of an ApsaraDB RDS for PostgreSQL instance
- Change the specifications of an ApsaraDB RDS for PPAS instance
- Change the specifications of an ApsaraDB RDS for MariaDB TX instance

Prerequisites

Your Alibaba Cloud account does not have overdue renewal orders.

Change items

<table>
<thead>
<tr>
<th>Change item</th>
<th>Description</th>
<th>Change method</th>
<th>Billing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edition</td>
<td>In MySQL 5.7, you can change your RDS edition from Basic to High-availability, or from High-availability to Enterprise. Your instance edition cannot be rolled back after the change is completed.</td>
<td>For more information, see Procedure.</td>
<td>For more information, see #unique_193.</td>
</tr>
<tr>
<td>Instance type</td>
<td>You can switch to any available instance type.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Issue: 20200702
<table>
<thead>
<tr>
<th>Change item</th>
<th>Description</th>
<th>Change method</th>
<th>Billing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage capacity</td>
<td>You can increase the storage capacity of all RDS instances. If your RDS instance uses subscription billing and local SSDs and you choose the <strong>change upon renewal</strong> method, the storage capacity can only be decreased.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
- The new storage capacity you specify must fall within the storage capacity range allowed for the instance type. For more information, see [#unique_82](#unique_82).
- If the RDS instance uses standard SSDs or enhanced SSDs (ESSDs), you cannot decrease its storage capacity.
- If the storage capacity range allowed for the instance type does not meet your needs, we recommend that you change the instance type.

| Storage type | When you change the ApsaraDB RDS for MySQL 5.7 instance edition from Basic to High-availability, the storage type of the instance can be changed from standard SSD to local SSD. | | |
### Change item

<table>
<thead>
<tr>
<th>Change item</th>
<th>Description</th>
<th>Change method</th>
<th>Billing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zone</td>
<td>You can migrate an instance across zones in the same region. The attributes, configurations, and endpoints of the instance remain unchanged after the migration. When you change the ApsaraDB RDS for MySQL 5.7 instance edition from High-availability to Enterprise, you must change the zone of the instance. For more information, see Procedure.</td>
<td>Migrate an ApsaraDB RDS MySQL instance across zones</td>
<td>Free of charge.</td>
</tr>
<tr>
<td>Primary/secondary switchover</td>
<td>A primary instance can be automatically or manually switched to a secondary instance.</td>
<td>Manually or automatically switch over services between the RDS MySQL master and slave instances</td>
<td></td>
</tr>
<tr>
<td>Network type</td>
<td>All instances can be deployed in VPCs, and some instances can also be deployed in the classic network. You can switch the network type of instances that support both network types.</td>
<td>Change the network type of an ApsaraDB RDS MySQL instance</td>
<td></td>
</tr>
<tr>
<td>VPC and VSwitch</td>
<td>You can change the VPC or VSwitch for some instances.</td>
<td>Switch to a new VPC and VSwitch for an RDS MySQL instance</td>
<td></td>
</tr>
<tr>
<td>Change item</td>
<td>Description</td>
<td>Change method</td>
<td>Billing</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maintenance window</td>
<td>You can change the maintenance window of an instance.</td>
<td>Set the maintenance window of an ApsaraDB RDS MySQL instance</td>
<td></td>
</tr>
<tr>
<td>Data replication</td>
<td>You can modify the data replication mode of an instance to improve database</td>
<td>Changes the data replication mode of an ApsaraDB RDS MySQL system</td>
<td></td>
</tr>
<tr>
<td>mode</td>
<td>availability.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instance parameter</td>
<td>You can modify instance parameters.</td>
<td>For more information, see Reconfigure the parameters of an ApsaraDB RDS for MySQL instance or Use a parameter template to manage parameters.</td>
<td></td>
</tr>
<tr>
<td>Engine version</td>
<td>The database engine version can only be upgraded from MySQL 5.5 to MySQL 5.6.</td>
<td>Upgrade the database engine version of an RDS MySQL instance</td>
<td></td>
</tr>
<tr>
<td>Change item</td>
<td>Description</td>
<td>Change method</td>
<td>Billing</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Billing method</td>
<td>The billing method of an instance can be switched between pay-as-you-go and subscription.</td>
<td><strong>Switch the billing method from pay-as-you-go to subscription</strong></td>
<td>For more information, see #unique_78.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Switch the billing method from subscription to pay-as-you-go</strong></td>
<td></td>
</tr>
<tr>
<td>Note:</td>
<td>You can only switch the billing method of a read-only instance from subscription to pay-as-you-go.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region</td>
<td>You cannot switch the region of an instance. Instead, you can create a new instance in the destination region and use Data Transmission Service (DTS) to migrate data from the source instance to the destination instance. After the data is migrated, you must modify the service endpoint. After your service is running normally, you can release the source instance.</td>
<td><strong>For more information, see Migrate data between ApsaraDB for RDS instances.</strong></td>
<td><strong>For more information, see Pricing, billing items, and billing methods and Pricing.</strong></td>
</tr>
</tbody>
</table>

**Note:**

- Specification changes do not change endpoints.
- You can create read-only instances to scale the read capability of your database system. For more information, see Create an ApsaraDB RDS for MySQL read-only instance.

**Precautions**

- Data may be migrated after you change specifications. After the migration is completed, the instance performs switchover during the scheduled time period. Incremental data is also synchronized during the switchover. A transient disconnection of about 30 seconds may occur, during which operations related to databases, accounts, and networks cannot be performed. We recommend that you change specifications within
the maintenance window and make sure that your application is configured with automatic reconnection policies.

- You do not need to manually restart the instance after the specifications are changed.
- You can only create a single primary RDS instance for a Basic Edition instance. No secondary instances are provided as hot backups. If the primary instance breaks down, is executing specification changes, or is upgrading its version, it may remain unavailable until the operations are done. If you require high service availability, we recommend that you use the High-availability, Cluster, or Enterprise Edition.

**Procedure**

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the instance resides.
3. Find the target RDS instance and click the instance ID.
4. In the Configuration Information section of the Basic Information page, click **Change Specifications**.
5. In the dialog box that appears, select a change method and click **Next**. This step is only required for subscription instances.
6. Change the specifications of the RDS instance. For more information, see Change items.

7. Specify the time to change the instance specifications.

   • **Switch Immediately After Data Migration**: The system executes the specification changes immediately after data migration is complete.
   
   • **Switch Within Maintenance Window**: The system executes the specification changes during the maintenance window you specify. For more information, see Set the maintenance window of an ApsaraDB RDS for MySQL instance.

**Note:**

If you want to change the maintenance window, follow these steps:

a. Click **Modify**.

b. In the **Configuration Information** section of the Basic Information page that appears, select a maintenance window and click **Save**.

c. Return to the Change Specifications page, refresh it, and continue to change specifications.

8. On the **Change Specifications** page, read and select Terms of Service, Service Level Agreement, and Terms of Use, and click **Confirm** to complete the payment.

**FAQ**

- How do I change the storage type of my RDS instance?

  For more information, see #unique_198
• Do I need to migrate data to a new instance if I only want to expand the storage capacity?

Check whether the storage capacity of the host where the RDS instance resides is sufficient. If yes, you can expand the storage capacity without migrating data. If no, you must migrate data to a new instance located on a host with sufficient storage capacity.

• Will the specifications of read-only instances be upgraded automatically after I upgrade the specifications of their primary RDS instance?

No, you must upgrade the specifications of read-only instances manually.

• Will my online services be interrupted while I change the specifications of my RDS instance?

While you change the specifications of your RDS instance, only a 30-second transient disconnection may occur.

• Will the endpoints and addresses of my RDS instance change after I change its specifications?

After you change the specifications of your RDS instance, its internal and public endpoints and read/write splitting endpoint remain unchanged, but the corresponding IP addresses may change. We recommend that you use the internal endpoint, public endpoint, or read/write splitting endpoint of your RDS instance to establish a connection from your application.

10.2 Manually or automatically switch over services between the RDS MySQL master and slave instances

This topic describes how to manually or automatically switch over services between the RDS MySQL master and slave instances. After the switchover, the master instance becomes the slave instance.

Prerequisites

The master instance is in the High-availability Edition.

Note:

RDS instances in the Basic Edition do not have slave instances, and therefore do not support service switchovers.
• Automatic switchover: the default switchover mode. When the master instance becomes faulty, your RDS services are automatically switched over to the slave instance.

• Manual switchover: You can manually switch over services between the master and slave instances even when the automatic switchover function is enabled.

**Note:**

In the High-availability edition, an RDS instance (referred to as the master instance) has a slave instance. Data is synchronized in real time between the master and slave instances. You can only access the master instance. The slave instance is used only as a backup and does not provide services. After a service switchover, the master instance is degraded to the slave instance.

**Precautions**

• Services may be disconnected during a switchover. Make sure that you configure automatic reconnection policies for your applications to avoid loss of services.

• If read-only RDS instances are mounted to your RDS instance, the data in the read-only RDS instances shows a few minutes' delay after a switchover. This is because it takes time to reestablish replication links and synchronize incremental data.

**Manually switch over services between the master and slave instances**

1. Log on to the RDS console.

2. In the upper-left corner, select the region where the target RDS instance is located.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click **Service Availability**.
5. In the **Availability Information** section, click **Switch Primary/Secondary Instance**.

6. Select the time at which you want to perform a switchover, and click **OK**.

**Note:**
During the switchover, you cannot perform operations such as managing databases and accounts and changing the network type. Therefore, we recommend that you select **Switch Within Maintenance Window**.

**Temporarily disable automatic switchover**

The automatic switchover function is enabled by default. In such case, services can be automatically switched over to the slave instance if the master instance becomes faulty. You can disable the automatic switchover function in one of the following situations:

- A sales promotion, during which you do not want a switchover to affect service availability.
- An important application upgrade, during which you do not want a switchover to incur any unexpected issues.
- A major event that requires stable system operation, during which you do not want a switchover to affect system stability.

1. Log on to the RDS console.

2. In the upper-left corner, select the region where the target RDS instance is located.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click Service Availability.

5. In the Availability Information section, click Configure Primary/Secondary Switchover.

   **Note:**
   If the Configure Primary/Secondary Switchover button is unavailable, make sure that the RDS instance is in the High-availability Edition.

6. Select Disable Temporarily, set Deadline, and click OK.

   **Note:**
   - When the specified Deadline arrives, the automatic switchover function is restored to enabled for the RDS instance.
By default, the automatic switchover function is disabled for one day. You can set the **Deadline** parameter to 23:59:59 seven days later at tops.

After the setting is complete, you can go to the **Service Availability** page to check the deadline for disabling the automatic switchover function.

10.3 Set the maintenance window of an ApsaraDB RDS MySQL instance

This topic describes how to set the maintenance window of an ApsaraDB RDS MySQL instance. To ensure database stability, the backend system performs maintenance operations on your RDS instance every day during the maintenance window you specify. The default maintenance window spans from 02:00 to 06:00 UTC+8. We recommend that you set the maintenance window to off-peak hours to avoid interference to your business.

**Precautions**

- Before the backend system starts maintenance, ApsaraDB for RDS sends notification emails to the contacts listed in your Alibaba Cloud account.
- To ensure smooth maintenance, your RDS instance enters the Instance Maintaining state prior to the maintenance window. While your RDS instance stays in Instance Maintainin...
g state, database access and query operations such as performance monitoring are still available. However, apart from account and database management and IP address whitelist configuration, all other modify operations such as upgrade, downgrade, and restart are temporarily unavailable.

- During the maintenance window, one or two transient disconnections may occur. Make sure that your application is configured to automatically reconnect to your RDS instance.

**Procedure**

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target RDS instance resides.
3. Find the target RDS instance. Then, click its ID, or click Manage in the Actions column.
4. In the Configuration Information section of the Basic Information page, click Configure next to Maintenance Window.
5. Select a maintenance window and click Save.

**Note:**
The maintenance window is in UTC+8.
**Related operations**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_199</td>
<td>Changes the maintenance window of an ApsaraDB for RDS instance.</td>
</tr>
</tbody>
</table>

**10.4 Migrate an ApsaraDB RDS MySQL instance across zones**

This topic describes how to migrate an ApsaraDB RDS MySQL instance across zones in the same region. After migration, all attributes, configurations, and endpoints of the instance remain unchanged. It takes several hours for a migration task to complete. The actual duration required for migration depends on the amount of data in an instance.

For more information about how to migrate ApsaraDB for RDS instances of other engines across zones, see the following topics:

- Migrate an ApsaraDB RDS SQL Server instance across zones
- Migrate an ApsaraDB RDS PostgreSQL instance across zones
- Migrate an ApsaraDB RDS PPAS instance across zones

**Precautions**

- Services are disconnected for 30 seconds during switchover. Make sure that your applications are configured with automatic reconnection policies.
- After you switch to a new VPC and VSwitch, the virtual IP addresses (VIPs) of the RDS instance change. Therefore, we recommend that you use an endpoint of the RDS instance to establish a connection with your application.
- The VIP changes temporarily interrupt the connection with Alibaba Cloud DRDS. You must update the endpoint information of your RDS instance in the DRDS console immediately after you change its VPC and VSwitch.
- The VIP changes temporarily interrupt the connections with Alibaba Cloud Data Management Service (DMS) and Data Transmission Service (DTS). The connections are automatically restored to normal.

**Migration types**

<table>
<thead>
<tr>
<th>Migration type</th>
<th>Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrate instances from one zone to another</td>
<td>The zone where the RDS instance is located is overloaded or cannot meet the performance requirements of the instance.</td>
</tr>
<tr>
<td>Migration type</td>
<td>Scenario</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Migrate instances from one zone to multiple zones</td>
<td>Disaster recovery across data centers is required to improve the disaster recovery capability of the instances. The primary and secondary instances are located in different zones. Compared with single-zone instances, multi-zone instances can withstand disasters at higher levels. For example, single-zone instances can tolerate server and rack faults, whereas multi-zone instances can tolerate data center faults.</td>
</tr>
<tr>
<td>Migrate instances from multiple zones to one zone</td>
<td>Specific features are required.</td>
</tr>
</tbody>
</table>

**Fees**

This feature is free of charge even if you migrate instances from one zone to multiple zones.

**Prerequisites**

- Your database system runs one of the following MySQL versions:
  - MySQL 8.0 (with local SSDs)
  - MySQL 5.7 (with local SSDs)
  - MySQL 5.6
  - MySQL 5.5
- The region where the instances are located contains multiple zones. For more information about regions and zones, see Regions and zones.
- The network connection mode is upgraded. For more information, see #unique_124.

**Procedure**

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click the instance ID.

4. On the Basic Information page, click **Migrate Across Zones**.

<table>
<thead>
<tr>
<th>Basic Information</th>
<th>Configure Whitelist to view the internal IP address.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance ID:</td>
<td></td>
</tr>
<tr>
<td>Region and Zone:</td>
<td>China (Hangzhou) ZoneH+ZoneI</td>
</tr>
<tr>
<td>Internal Endpoint:</td>
<td>Configure Whitelist</td>
</tr>
<tr>
<td>Public Endpoint:</td>
<td>Apply for Public Endpoint</td>
</tr>
<tr>
<td>Storage Type:</td>
<td>Local SSD</td>
</tr>
</tbody>
</table>

5. In the dialog box that appears, specify the destination zone, VSwitch, and switching time, and then click **OK**.

After you click **OK**, the system copies data to the destination zone while the instance runs normally. After data is copied, services are switched to the new connection at the time that you specified (**Switch Now** or **Switch Within Maintenance Window**).

**Note:**

- Services are disconnected for 30 seconds during switchover. Make sure that your applications are configured with automatic reconnection policies. Otherwise, you must manually reconnect your applications to RDS.
- Some services may be switched after 10 minutes because the DNS cache is not immediately flushed, which causes a disconnection again.
If you need to change the maintenance window, perform the following operations:

a. Click **Change**.

b. In the **Configuration Information** section, select a maintenance window and click **Save**.

c. Return to the Basic Information page, refresh it, and perform the migration again.

**Related operations**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_203</td>
<td>Migrates an RDS instance from one zone to another.</td>
</tr>
</tbody>
</table>

**10.5 Changes the data replication mode of an ApsaraDB RDS MySQL system**

This topic describes how to change the mode of data replication between a primary ApsaraDB RDS MySQL instance and its secondary instances to increase database availability.

**Data replication modes**

- Synchronous
  - After a primary instance completes the updates initiated from your application, the instance synchronously replicates update logs to its secondary instances. The
system considers the update transactions submitted only after at least one secondary instance receives and stores the logs.

- In synchronous mode, data keeps being replicated synchronously no matter what the circumstances are.

- The synchronous mode is only available when the database system consists of three or more instances. This requires your database system to be in the Enterprise Edition. Additionally, the Enterprise Edition only supports the synchronous mode.

• Semi-synchronous

After a primary instance completes the updates initiated from your application, the instance synchronously replicates update logs to its secondary instance. After the secondary instance receives the logs, the database system considers the update transactions submitted, no matter whether the secondary instance finishes executing the updates specified in the logs.

If the secondary instance becomes unavailable or the primary and secondary instances cannot communicate properly, the data replication mode is downgraded from semi-synchronous to asynchronous.

• Asynchronous

After a primary instance completes the updates initiated from your application, the instance responds to your application immediately while starting to asynchronously replicate update logs to its secondary instance. Such updates include add, delete, and modify operations. In asynchronous mode, you can still perform operations on the primary instance even when the secondary instance is unavailable. Additionally, unavailability of the primary instance rarely results in data inconsistencies between the primary and secondary instances.

**Prerequisites**

Your database system runs one of the following MySQL versions and RDS editions:

• MySQL 8.0 High-availability Edition (with local SSDs)
• MySQL 5.7 High-availability Edition (with local SSDs)
• MySQL 5.6 High-availability Edition
• MySQL 5.5

**Procedure**

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID.

4. In the left-side navigation pane, click **Service Availability**.

5. Click **Change Data Replication Mode**.

6. In the dialog box that appears, select a data replication mode and click **OK**.

### Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_204</td>
<td>Changes the data replication mode and high availability mode of an ApsaraDB for RDS instance.</td>
</tr>
</tbody>
</table>
10.6 Switch the billing method from pay-as-you-go to subscription

This topic describes how to switch the billing method of an ApsaraDB RDS for MySQL instance from pay-as-you-go to subscription.

Impacts

Changing billing methods will not impact the performance of your cluster.

Precautions

• You cannot upgrade the specifications of a subscription-based instance if the purchase order of the instance has not been paid. Otherwise, the unpaid order will be invalid. You need to cancel this order on the Billing Management page and change the billing method of the instance to subscription.

Prerequisites

• The type of the instance is available for purchase. For more information about phased-out instance types, see Primary instance types. If you need to change the billing method of a phased-out instance type to subscription, change the instance type first. For more information, see Change specifications.
• The billing method of the instance is pay-as-you-go.
• The instance must be in the Running state.
• The instance has no unpaid subscription orders.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target instance and go to the **Switch to Subscription Billing** page by using one of the following methods:

   - In the **Actions** column corresponding to the instance, click **Switch to Subscription Billing**.
   - Click the instance ID to go to the Basic Information page. In the **Status** section, click **Switch to Subscription Billing**, as shown in the following figure.

![Switch to Subscription Billing](image)

4. Specify **Duration** and check service agreement.

5. Click **Pay Now**.

   **Note:**
   You must complete the subscription order generated for the instance. You cannot purchase a new instance or change the billing method of another instance until you pay for or cancel this order. You can pay for or cancel this order on the **Billing Management** page.

6. Pay the order as prompted.

**Related operations**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_205</td>
<td>Changes the billing method of an ApsaraDB for RDS instance.</td>
</tr>
</tbody>
</table>

**10.7 Switch the billing method from subscription to pay-as-you-go**

This topic describes how to switch the billing method of an ApsaraDB for RDS instance from subscription to pay-as-you-go.

**Prerequisites**

- The billing method of the instance is subscription. For more information about the billing methods, see #unique_78.
- The instance is in the Running state.
Billing

After you change the billing method of the instance from subscription to pay-as-you-go, a refund (if applicable) will be returned to your original payment account.

Refund = (Remaining days/Total days) × Order cash amount × 70%

**Note:**
The order cash amount indicates the actual amount paid in cash, excluding vouchers or coupons.

Precautions

Changing the billing method of an instance will not affect its performance.

**Note:**
For long-term use, we recommend that you select the subscription billing method because it is more cost-effective than the pay-as-you-go billing method. The longer the subscription period, the larger the discount will be.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target ApsaraDB for RDS instance is located.
3. Find the target RDS instance and click the instance ID.
4. On the **Basic Information** page of the instance, find the Status section and click **Switch to Pay-as-you-go Billing** in the upper-right corner of the section.

5. Confirm the instance information, select the terms of service, and then click **Pay Now** to complete the payment.

**Related operation**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_205</td>
<td>Changes the billing method of an ApsaraDB for RDS instance.</td>
</tr>
</tbody>
</table>

### 10.8 Change the network type of an ApsaraDB RDS MySQL instance

This topic describes how to change the network type of an ApsaraDB RDS MySQL instance between Classic Network and VPC.

For more information about how to change the network type in other database engines, see the following topics:

- Change the network type of an RDS SQL Server instance
- Change the network type of an RDS PostgreSQL instance
- Change the network type of an RDS PPAS instance

**Network types**

- **Classic Network**: RDS instances in a classic network are not isolated. You can only use whitelists to block unauthorized access to your RDS instance.
- **VPC**: Each Virtual Private Cloud (VPC) is an isolated network. We recommend that you choose the VPC network type because it is more secure.

You can customize the routing table, IP address range, and gateway in a VPC. To migrate applications to the cloud without interruption, you can connect your own data center to a VPC in the cloud over a private connection or VPN to build a virtual data center.

**Note:**
• You can choose the Classic Network or VPC network type and switch between them for free.
• Before you change the network type of an ApsaraDB RDS MySQL instance, you must change its network isolation mode to enhanced whitelist. For more information, see Switch to the enhanced whitelist mode for an RDS MySQL instance.

**Change the network type from VPC to Classic Network**

**Precautions**

• After the network type is changed, the internal endpoint of the RDS instance remains unchanged, but the IP address associated with the internal endpoint changes.
• After the network type is changed, an ECS instance located in the same VPC as the RDS instance can no longer connect to the RDS instance by using the internal endpoint. Make sure that you update the endpoint information on the ECS instance.
• While you change the network type, a 30-second transient disconnection may occur. To avoid interference to your business, we recommend that you change the network type of the RDS instance during off-peak hours, or make sure that your application is configured to automatically reconnect to the RDS instance.
• If the RDS instance runs MySQL 5.7 or MySQL 8.0 and is in the High-availability Edition with standard SSDs or enhanced SSDs (ESSDs), it does not support the Classic Network type. Therefore, you cannot change its network type from VPC to Classic Network.

**Procedure**

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click **Database Connection**.
5. In the Database Connection section, click **Switch to Classic Network**.

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

   After the network type is changed to Classic Network, only an ECS instance located in the classic network can connect to the RDS instance by using the internal endpoint. Make sure that you add the internal endpoint of the RDS instance to the ECS instance.

7. Configure a whitelist to enable the ECS instance to connect to the RDS instance by using the internal endpoint.

   - If the RDS instance works in standard whitelist mode, add the private IP address of the ECS instance to any IP address whitelist.

   ![Whitelist Settings](image)

   - If the RDS instance works in enhanced whitelist mode, add the private IP address of the ECS instance to an IP address whitelist of the Classic Network type. For more
information, see Switch to the enhanced whitelist mode for an RDS MySQL instance. If no IP address whitelists of the Classic Network type are available, create one.

**Change the network type from Classic Network to VPC**

**Procedure**

1. Log on to the ApsaraDB for RDS console.

2. In the upper-left corner of the page, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID.

4. In the left-side navigation pane, click *Database Connection*.

5. Click *Switch to VPC*.

6. In the dialog box that appears, select a VPC and a VSwitch, and specify whether to retain the endpoints used in the classic network.
   - Select a VPC. We recommend that you select the VPC that houses your ECS instance. If you do not perform this procedure, your ECS instance cannot connect to the RDS instance over an internal network unless you create a connection by using Express
Connect or VPN Gateway between the VPCs of the ECS and RDS instances. For more information, see ClassicLink and Establish a connection between two VPCs.

- Select a VSwitch. If there are no VSwitches available in the selected VPC, create one in the same zone where the RDS instance resides. For more information, see Create a VSwitch.

```markdown
Switch to classic, include endpoint(s):
Internal Port: -8080 -3306 -3306  rds.aliyun.com

Switch to:
VPC: ..............................
Virtual Switch: ..............................

If the switch you need is not in the list, please create a new switch first on the VPC console.

Note: Switching to Virtual Private Cloud (VPC) will cause an intermittent interruption, and the ECS in the classic network will not be able to access the database. If you need to reserve the intranet address of the classic network, check the following option.
```

- Clear or select Reserve original classic network. The following table describes the different operations.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
</table>
| Clear the Reserve original classic network option | The endpoints used in the classic network are replaced with those to be used in the selected VPC.  
  While you change the network type, a 30-second transient disconnection may occur, and the connection from the ECS instance in the classic network to the RDS instance is closed. |
<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the Reserve original classic network option</td>
<td>The endpoints used in the classic network are retained, and the new endpoints you want to use in the selected VPC are generated. In the example shown in the following figure, <strong>hybrid access mode</strong> is enabled. This means that ECS instances in both the classic network and the selected VPC can connect to the RDS instance by using the internal endpoint. While you change the network type, the RDS instance remains connected, and the connection from the ECS instance in the classic network to the RDS instance remains available until the endpoints used in the classic network expire. Before the endpoints used in the classic network expire, you must add the new endpoints you want to use in the selected VPC to the ECS instance. This helps migrate your business to the VPC without interruption. Within seven days before the endpoints used in the classic network expire, the system sends Short Message Service (SMS) messages to the mobile phone bound to your Alibaba Cloud account every day.</td>
</tr>
</tbody>
</table>

![Database Connection](#)  
**Note:** Use the preceding connection string to connect to the instance. You need to change the VIP in the connection string.  

![Original classic endpoint](#)  
For more information, see [Configure a hybrid access solution to smoothly migrate an RDS instance from the classic network to a VPC](#).  

7. Add the private IP address of the ECS instance in the selected VPC to an IP address whitelist of the VPC type. This enables the ECS instance to connect to the RDS instance.
over an internal network. If no IP address whitelists of the VPC type are available, create one.

![Network isolation mode: enhanced whitelist](image)

<table>
<thead>
<tr>
<th>Whitelist Settings</th>
<th>SQL Audit</th>
<th>SSL Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network isolation mode: enhanced whitelist</td>
<td>Each of the following whitelists serve a classic network or VPC.</td>
<td></td>
</tr>
</tbody>
</table>

8. Add the new endpoints of the RDS instance to the target ECS instance.

- If you have selected the Reserve original classic network option, add the new endpoints to the ECS instance before the endpoints used in the classic network expire.
- If you have cleared the Reserve original classic network option, add the new endpoints to the ECS instance immediately after the network type change is complete.

Note:

If you want to connect an ECS instance in the classic network to an RDS instance in a VPC over an internal network, you must use ClassicLink to establish a connection, or migrate the ECS instance to the same VPC as the RDS instance.

FAQ

How do I change the VPC and VSwitch of an ApsaraDB RDS MySQL instance?

- You can only directly change the VPC and VSwitch if the instance supports the change. For more information, see Switch to a new VPC and VSwitch for an RDS MySQL instance.
- If the instance supports network type changes between VPC and Classic Network:
  1. Change the network type from VPC to Classic Network.
  2. Change the network type from Classic Network to VPC and select the target VPC.
- If the instance does not support network type changes between VPC and Classic Network:

  Purchase a new instance and make sure that you select the target VPC. Then, migrate data to the new instance. For more information, see Migrate data between ApsaraDB for RDS instances.
10.9 Switch to a new VPC and VSwitch for an RDS MySQL instance

This topic describes how to switch to a new VPC and VSwitch for an RDS MySQL instance.

Context

- For local SSD instances, you can modify Virtual Private Cloud and vSwitch directly.
- For standard SSD or ESSD instances, you can only modify the vSwitch, not the Virtual Private Cloud.

**Note:**
For more information about how to switch between the PostgreSQL engine Virtual Private Cloud and vSwitch, see #unique_215.

Impact

- The RDS instance is disconnected for about 30 seconds during the switchover. Make sure that your application can automatically reconnect to the RDS instance.
- The virtual IP address of the RDS instance changes after the switchover. We recommend that you use an endpoint of the RDS instance to establish a connection.
- The DRDS is unavailable for a short period of time due to the change to the virtual IP address during the switchover. You must update and view the connection information of the RDS instance in the console immediately after the switchover is complete.
- The DMS and DTS is unavailable for a short period of time due to the change to the virtual IP address during the switchover, and automatically returns to normal after the switchover is complete.
- You must clear the cache on your database client properly. If you do not do so, data can be read but cannot be written into the RDS instance.

Procedure

1. Log on to the RDS console.

### Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_214</td>
<td>Changes the network type of an ApsaraDB for RDS instance.</td>
</tr>
</tbody>
</table>
2. In the upper-left corner, select the region where the target RDS instance is located.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click **Database Connection**.

5. In the upper-right corner, click **Switch VSwitch**.

6. Select a VPC and a VSwitch, then click **OK**.

**Note:**
To create a VPC or VSwitch, click **go to the VPC console.**

7. In the displayed dialog box, click the button for confirming the switchover.

FAQ

How do I change the VPC of an RDS instance if that instance does not allow me to switch to a new VPC and VSwitch?

- If your RDS instance supports direct network type changes:
  1. Switch the network type from VPC to Classic Network.
  2. Switch the network type from classic network to the target VPC.

  **Note:**
  For more information, see [Change the network type of an ApsaraDB RDS MySQL instance](#).

- If your RDS instance does not support direct network type changes:

  Purchase a new RDS instance with the destination VPC selected, and then migrate data to the new RDS instance. For more information, see [Migrate data between ApsaraDB for RDS instances](#).
For instances that do not support switching Virtual Private Cloud and vswitches directly, see #unique_216.
11 Version upgrade

11.1 Update the kernel version of an ApsaraDB RDS for MySQL instance.

ApsaraDB RDS for MySQL supports automatic and manual updates of the kernel version. These updates increase performance, unveil new features, and fix known issues.

For more information about the features that are provided by various kernel versions of ApsaraDB RDS for MySQL, see #unique_218.

For more information about updating the kernel version of an ApsaraDB RDS for PostgreSQL instance, see Updating the kernel version of an ApsaraDB RDS for PostgreSQL instance.

Note:

ApsaraDB for RDS instances running SQL Server, PPAS, or MariaDB do not support kernel version updates.

Introduction

RDS for MySQL uses the automatic kernel version update mode by default. You can log on to the ApsaraDB for RDS console, navigate to the Basic Information page of your RDS instance, and then in the Configuration Information section view the current Minor Version Upgrade Mode.

- **Auto**: When a new kernel version is released, Alibaba Cloud pushes a notification to you and automatically updates the kernel version of your RDS instance during the specified maintenance window.
• **Manual**: When the lifecycle of the kernel version installed on your RDS instance ends, Alibaba Cloud pushes a notification to you. The notification states that you must update your RDS instance to the latest stable kernel version within one month. In most cases, the lifecycle of a kernel version spans one year. You can manually update the kernel version on the Basic Information page.

**Note:**
A kernel version is deprecated after its lifecycle ends.

**Precautions**
- A kernel version update will restart your RDS instance. This may incur a 30-second brief disconnection. We recommend that you perform the update during off-peak hours or make sure that your application is configured to automatically reconnect to your RDS instance.
- A kernel version update cannot be rolled back.
- A kernel version update is triggered when you upgrade your RDS instance.

**Configure the kernel version update mode**

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the Configuration Information section of the Basic Information page, click Configure to the right of Minor Version Upgrade Mode.
5. Select **Auto** or **Manual** and click **OK**.

Manually update the kernel version

1. Log on to the ApsaraDB for RDS console.

2. In the top navigation bar, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID.

4. In the Configuration Information section of the **Basic Information** page, click **Upgrade Minor Version**.
5. In the dialog box that appears, specify the update time and click **OK**.

![Upgrade Kernel Version Settings](image)

- **Current Version**: rds_20200110
- **Available Upgrade**: rds_20200331
  - To compare differences between versions, see [reference documentation](#).
- **Upgrade Time**:  
  - Upgrade Immediate
  - Upgrade within maintenance period (Current Setting:02:00-06:00 [Modify](#))

Note: Upgrading the minor version will restart the RDS instance and cause a disconnection of 30 seconds. We recommend that you upgrade the minor version during off-peak hours. Ensure that your applications are configured with automatic reconnection policies. Otherwise, you must manually reconnect your applications to RDS.

**FAQ**

- After I updated the kernel version of my RDS instance, why does the SELECT @@version statement still return the source kernel version that I used before the update?

  The kernel version that you updated is the kernel version of Alibaba Cloud instead of the kernel version of MySQL. You need to run the `show variables like '%rds_release_date%'` command to view the kernel version of your RDS instance.

- Is each update targeted only for the next kernel version?

  No, each update is targeted for the latest kernel version.

**Related operations**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_221</td>
<td>Updates the kernel version of an ApsaraDB RDS for MySQL instance.</td>
</tr>
</tbody>
</table>
11.2 Upgrade the database engine version of an RDS MySQL instance

This topic describes how to upgrade the database engine version of an ApsaraDB RDS MySQL instance.

For more information about how to upgrade the versions of other database engines, see the following topics:

- Upgrade from Basic Edition to High-availability Edition
- Upgrade SQL Server 2008 R2 to SQL Server 2012 or SQL Server 2016
- Upgrade an instance from SQL Server 2012 to SQL Server 2016

Precautions

- The database engine version cannot be downgraded.
- The database engine version can only be upgraded from MySQL 5.5 to MySQL 5.6.
- Before performing a database engine version upgrade, we recommend that you purchase an Apsaradb RDS MySQL instance of the target version to test the version compatibility.
- During a database engine version upgrade, your ApsaraDB RDS MySQL instance may encounter a 30-second transient disconnection. We recommend that you perform the upgrade during off-peak hours or make sure that your application can automatically reconnect to your Apsaradb RDS MySQL instance.

Procedure

1. Log on to the Apsaradb for RDS console.
2. In the upper-left corner of the console, select the region where the target RDS instance resides.
3. Find the target RDS instance and click the instance ID.
4. In the Configuration Information section of the Basic Information page, click **Upgrade Database**.

<table>
<thead>
<tr>
<th>Configuration Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Family: General</td>
</tr>
<tr>
<td>Database Memory: 1024MB</td>
</tr>
<tr>
<td>Time Segment: 02:00-06:00</td>
</tr>
</tbody>
</table>

5. In the dialog box that appears, select the target version and click **Upgrade**.

**Related operations**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_225</td>
<td>Upgrades the database engine version of an ApsaraDB for RDS instance.</td>
</tr>
</tbody>
</table>
12 Instance parameters

12.1 Reconfigure the parameters of an ApsaraDB RDS for MySQL instance

This topic describes how to view and reconfigure the parameters of an ApsaraDB RDS for MySQL instance by using the ApsaraDB for RDS console or the API. You can also view the parameter reconfiguration history in the ApsaraDB for RDS console.

Precautions

- To ensure instance stability, the ApsaraDB for RDS console allows you to reconfigure only some parameters. If you cannot find the parameter that you want to reconfigure, submit a ticket.
- When you reconfigure parameters, you can refer to the Value Range column on the Editable Parameters tab in the ApsaraDB for RDS console.
- The new values of some parameters can take effect only after an instance restart. For more information, refer to the Force Restart column on the Editable Parameters tab in the ApsaraDB for RDS console. We recommend that you reconfigure parameters during off-peak hours and make sure that your application is configured to automatically reconnect to your RDS instance.

Reconfigure parameters

Note:

If you want to reconfigure a number of parameters at a time, we recommend that you use a parameter template. For more information, see Use a parameter template to manage parameters.

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID.

4. In the left-side navigation pane, click Parameters.

5. Perform the following operations based on your business scenario:
   - Apply a parameter template to the RDS instance.

   **Note:**
   If an instance restart is required, we recommend that you apply the parameter template during off-peak hours and make sure that your application is configured to automatically reconnect to the RDS instance.

   a. On the Editable Parameters tab, click **Apply Template**.
   b. In the Apply Template dialog box, select the parameter template and click **OK**.
- You can view the number of parameters in the parameter template and check whether you need to restart the RDS instance. For more information about the parameter reconfiguration history, click **View Parameter Change**.

- If you cannot find the parameter template, check whether the parameter template belongs to the same region as the RDS instance. If they belong to different regions, you can copy the parameter template to the region to which
ApsaraDB for RDS

RDS MySQL Database / 12 Instance parameters

the RDS instance belongs. For more information, see Use a parameter template to manage parameters.

Apply Template

<table>
<thead>
<tr>
<th>Template Name:</th>
<th>template2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Parameters:</td>
<td>3</td>
</tr>
<tr>
<td>Force Restart:</td>
<td>Yes</td>
</tr>
<tr>
<td>Update Time:</td>
<td>2019-08-30 11:02:18</td>
</tr>
</tbody>
</table>

View Parameter Change

- Export the parameter settings of the RDS instance as a parameter template in the current region.

  a. On the Editable Parameters tab, click Export as Template.
  
  b. Configure the following parameters.

Export as Template

| Template Name: | template_test |

The template name must be 8 to 64 characters in length and can contain letters, digits, periods (.), and underscores (_). It must start with a letter.

description:

The description must be 0 to 200 characters in length. It can be in any language.
### ApsaraDB for RDS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template Name</td>
<td>The name of the parameter template. The name must be 8 to 64 characters in length. It can contain letters, digits, periods (.), and underscores (_). It must start with a letter.</td>
</tr>
<tr>
<td>Description</td>
<td>The description that helps identify the parameter template. The description can be up to 200 characters in length.</td>
</tr>
</tbody>
</table>

- **c.** Click **OK**.

- Export the parameter settings of the RDS instance to your computer.

  On the Editable Parameters tab, click **Export Parameters**. The parameter settings of the RDS instance are exported as a TXT file to your computer.

  - **a.** Reconfigure the parameters in the TXT file, click **Import Parameters**, and in the Import Parameters dialog box paste the new parameter values you have copied from the TXT file.

  - **b.** Click **OK**.

  - **c.** In the upper-right corner of the page, click **Apply Changes**.

**Note:**

- If the new parameter values can take effect only after an instance restart, the system prompts you to restart the RDS instance. We recommend that you restart the RDS instance during off-peak hours and make sure that your application is configured to automatically reconnect to the RDS instance.
- Before the new parameter values are applied, you can click **Cancel Changes** to cancel them.

<table>
<thead>
<tr>
<th>Actual Value</th>
<th>Force Restart</th>
<th>Value Range</th>
<th>Parameter Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>No</td>
<td>[ON</td>
<td>OFF]</td>
</tr>
<tr>
<td>1</td>
<td>No</td>
<td>[1-65535]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>No</td>
<td>[1-65535]</td>
<td></td>
</tr>
<tr>
<td>OFF</td>
<td>No</td>
<td>[ON</td>
<td>OFF]</td>
</tr>
</tbody>
</table>

- Reconfigure a single parameter of the RDS instance.
  
  a. On the **Editable Parameters** tab, find the parameter that you want to reconfigure, and click the icon in the **Actual Value** column.
  
  b. Enter a new value based on the prompted value range.

  ![Input Field](image)

  c. Click **Confirm**.
  
  d. In the upper-right corner of the page, click **Apply Changes**.

  **Note:**
  
  - If the new parameter value can take effect only after an instance restart, the system prompts you to restart the RDS instance. We recommend that you restart the RDS instance during off-peak hours and make sure that your application is configured to automatically reconnect to the RDS instance.
  
  - Before the new parameter value is applied, you can click **Cancel Changes** to cancel it.

**View the parameter reconfiguration history**

1. Log on to the **ApsaraDB for RDS console**.
2. In the top navigation bar, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID.

4. In the left-side navigation pane, click **Parameters**.

5. Click the **Edit History** tab.

6. Select a time range and click **Search**.

**FAQ**

- **After I reconfigure the parameters of my RDS instance, will the new parameter values take effect immediately? And do I need to restart my RDS instance?**

  Some parameters do not require an instance restart after they are reconfigured. The new values of these parameters take effect in about 5 minutes. Some parameters require an instance restart after they are reconfigured. For more information, refer to the **Force Restart** column on the **Editable Parameters** tab in the ApsaraDB for RDS console.

- **After I reconfigure the parameters of my RDS instance, why cannot the new parameter values take effect?**

  After you reconfigure the parameters of your RDS instance, you must click **Apply Changes** in the upper-right corner of the Editable Parameters tab in the ApsaraDB for RDS console. This makes the new parameter values take effect.
12.2 Use a parameter template to manage parameters

This topic describes how to manage the parameters of an ApsaraDB for RDS instance by using a parameter template. ApsaraDB RDS for MySQL instances provide system parameter templates and custom parameter templates.

Prerequisites

Instances must run one of the following MySQL versions:

- MySQL 8.0
- MySQL 5.7
- MySQL 5.6

Context

To ensure service availability, you can only configure certain parameters from the ApsaraDB RDS for MySQL console. ApsaraDB RDS for MySQL provides various system parameter templates and allows you to customize parameter templates to suit your business requirements in high-performance and other scenarios.

Note:

For more information about how to configure a single parameter, see Reconfigure the parameters of an ApsaraDB RDS for MySQL instance.

System template introduction

Note:

System templates are only provided for MySQL High-availability instances based on local SSDs. For AsparaDB RDS for MySQL instances of other editions, you can Create a custom parameter template.
ApsaraDB RDS for MySQL High-availability instances based on local SSDs can use the following system parameter templates:

- Default parameter template

  It ensures the highest data security but costs more time. Data is semi-synchronously replicated. The parameters are specified with the following values to protect data:

  - InnoDB
    - `innodb_flush_log_at_trx_commit = 1`
    - `sync_binlog = 1`
  - X-Engine (only the default parameter template is provided)
    - `sync_binlog = 1`

- Asynchronous parameter template

  It ensures high data security and fast speed. Data is asynchronously replicated. The parameters are specified with the following values to protect data:

  - `innodb_flush_log_at_trx_commit = 1`
  - `sync_binlog = 1`

- High-performance parameter template

  It ensures common data security but costs less time. Data is asynchronously replicated. The parameters are specified with the following values to protect data:

  - `innodb_flush_log_at_trx_commit = 2`
  - `sync_binlog = 1000`

**Note:**

You cannot modify parameters in the system parameter templates by using a custom parameter template.

The following table describes the parameters in the system parameter templates:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>innodb_flush_log_at_trx_commit</code></td>
<td>1</td>
<td>When you submit a transaction, the system writes the transaction log from the buffer to the log file and synchronizes the log file to the disk immediately.</td>
</tr>
</tbody>
</table>
### Parameter Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>When you submit a transaction, the system writes the transaction log from</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the buffer to the log file but does not synchronize the log file to the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>disk immediately. The log file is written to the disk every second. If the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>system stops responding before a write operation is performed, logs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>generated within the last second will be lost.</td>
</tr>
<tr>
<td>sync_binlog</td>
<td>1</td>
<td>When you submit a transaction, the binary log file is written to the disk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and the disk is immediately refreshed. The log file is not written to the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>buffer.</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>The buffer is written to the disk and the disk is refreshed whenever</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,000 records are submitted to the buffer. This may result in data loss.</td>
</tr>
</tbody>
</table>

### Apply a Parameter Template

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target RDS instance resides.
3. In the left-side navigation pane, select Parameter Templates.
4. In the Custom Parameter Templates or System Parameter Templates tab, find the template that you want to apply.
5. Click Apply to Instance in the Actions column.
6. Select RDS instances from the All Instances section, click to move them to the Selected Instances section, and check the parameter comparison.

**Note:**
Before you apply a parameter template to multiple RDS instances, you must verify that the parameters are suitable for the RDS instances.

7. Click OK.
Create a custom parameter template

1. Log on to the ApsaraDB for RDS console.

2. In the upper-left corner of the page, select the region where the target RDS instance resides.

3. In the left-side navigation pane, click Parameter Templates. In the upper-right corner, click Create Parameter Template.

4. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template Name</td>
<td>Enter the name of the parameter template. The name must be 8 to 64 characters in length and can contain letters, digits, periods (.), and underscores (_). It must start with a letter.</td>
</tr>
<tr>
<td>Database Engine</td>
<td>Set the value to MySQL.</td>
</tr>
<tr>
<td>Engine Version</td>
<td>Valid values: MySQL 5.6, 5.7, and 8.0.</td>
</tr>
<tr>
<td>Description</td>
<td>Enter a description that helps identify the parameter template. The description can be up to 200 characters in length.</td>
</tr>
</tbody>
</table>
### Add Parameter

Click Add Parameter and select a parameter from the Parameter drop-down list. Then, you can set the parameter value as well as view the value range and default value.

**Note:**

- For more information about available parameters, see Editable Parameters on the Parameters page.
- If you want to add another parameter, you must click Add Parameter again.
- If you want to remove a parameter, you must click Delete to the right of the parameter.

### Import

If you have exported a parameter template to your computer, you can edit that template based on your business needs and then click Import to copy its parameters. For more information about how to export a parameter template, see Reconfigure the parameters of an ApsaraDB RDS for MySQL instance.

5. Click Confirm.

### Clone a parameter template

You can clone a parameter template from the current region to another region.

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target RDS instance resides.
3. In the left-side navigation pane, select **Parameter Templates**.

4. Find the target parameter template, and click **Clone** in the Actions column.

5. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region</td>
<td>Select the destination region to which you want to clone the parameter template.</td>
</tr>
<tr>
<td>Template Name</td>
<td>Enter the name of the parameter template. The name must be 8 to 64 characters in length. It can contain letters, digits, periods (.), and underscores (_). It must start with a letter.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Description</td>
<td>Enter a description that helps identify the parameter template. The description can be up to 200 characters in length.</td>
</tr>
</tbody>
</table>

6. Click **OK**.

**Manage parameter templates**

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target RDS instance resides.

3. In the left-side navigation pane, click **Parameter Templates**.

4. Manage parameter templates in this region.

**Note:**
You can only perform the **View** and **Apply to Instance** operations on system templates.

To view a parameter template, follow these steps:

Find the target parameter template, and click **View** in the Actions column. On the page that appears, view the basic information and parameters of the template.

To modify a parameter template, follow these steps:

a) Find the target parameter template, and click **Edit** in the Actions column. For more information, see Step 4 in the "Create a parameter template" section.

b) Click **Edit**.

To delete a parameter template, follow these steps:

Find the target parameter template, and click **Delete**. In the message that appears, click **OK**.

---

**Note:**

Deleting a parameter template will not affect the RDS instances to which the template was applied.

---

### Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_230</td>
<td>Creates an ApsaraDB for RDS parameter template.</td>
</tr>
<tr>
<td>#unique_231</td>
<td>Modifies an ApsaraDB for RDS parameter template.</td>
</tr>
<tr>
<td>#unique_232</td>
<td>Clones an ApsaraDB for RDS parameter template from one region to another.</td>
</tr>
<tr>
<td>#unique_233</td>
<td>Queries the ApsaraDB for RDS parameter templates available to a region.</td>
</tr>
<tr>
<td>#unique_234</td>
<td>Queries an ApsaraDB for RDS parameter template.</td>
</tr>
<tr>
<td>Operation</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>#unique_235</td>
<td>Deletes an ApsaraDB for RDS parameter template.</td>
</tr>
</tbody>
</table>
13 Backup

13.1 Back up an ApsaraDB RDS for MySQL instance

ApsaraDB RDS for MySQL supports both automatic and manual backups. If the data of your RDS instance is lost or corrupted, you can restore the instance by using its backups. If your RDS instance is used with local SSDs, you can also retain its backups for a specific period of time after you release the instance.

For more information about how to back up your RDS instance in other database engines, see the following topics:

- #unique_237
- #unique_238
- #unique_239
- #unique_240

**Note:**

This topic describes the default backup function, which stores backups to the region where your RDS instance resides. You can also store backups to a different region. For more information, see Back up an ApsaraDB RDS for MySQL instance across regions.

**Features**

If your RDS instance is used with local SSDs, you can retain backups for a long period of time. In addition, you can specify a policy that is used to retain backups after you release the instance. This allows you to avoid data losses caused by unintended operations.

**Billing**

Each RDS instance is allocated a quota of free backup space. If your usage exceeds the capacity of the quota, you must pay for the extra backup space that you use. We recommend that you specify a backup cycle that best meets your business needs and maximizes the utilization of the free backup space. For more information about the quota of free backup space, see View the quota of free backup space for an ApsaraDB RDS for MySQL instance.

After the quota of free backup space is exhausted, your backups are charged based on the specified retention period. For more information about the pricing of regular backups...
that are retained for 730 days or less, visit ApsaraDB RDS for MySQL pricing. For more information about the pricing of archived backups that are retained for more than 730 days, see Pricing of Long-Term Backups on Alibaba Cloud International Site.

Precautions

- Your RDS instance supports archived backups only when it is used with local SSDs.
- Do not execute data definition language (DDL) statements during a backup. If you do so, the backup may fail due to table locks.
- We recommend that you back up your RDS instance during off-peak hours.
- If the data volume is large, backing up your RDS instance may require a long time.
- Backups are retained based on the specified retention period. We recommend that you download the backups you require to your computer before they are deleted.
- If the number of tables created on your RDS instance exceeds 50,000, you cannot restore individual databases or tables. If the number of tables exceeds 600,000, you cannot back up your RDS instance. In these cases, we recommend that you shard the databases on your RDS instance.
### Overview of data and log backups

<table>
<thead>
<tr>
<th>Data backup</th>
<th>Log backup</th>
</tr>
</thead>
<tbody>
<tr>
<td>You can back up the data of your RDS instance. This includes physical backup, logical backup, and snapshot backup. The data backup files can be used to restore your RDS instance. Your RDS instance performs physical or snapshot backup based on the storage media that you use:</td>
<td>You can back up the binary logs of your RDS instance. The binary log files can be used to restore your RDS instance to a point in time. Your RDS instance automatically backs up its binary logs by default.</td>
</tr>
<tr>
<td>• MySQL 5.5, MySQL 5.6, MySQL 5.7, and MySQL 8.0 with local SSDs (in the High-availability or Enterprise Edition):</td>
<td>Note:</td>
</tr>
<tr>
<td>- Support full physical backup when the automatic backup mode is enabled.</td>
<td>• Binary log files occupy disk space on your RDS instance.</td>
</tr>
<tr>
<td>- Support full physical backup, full logical backup, and single-database logical backup when the manual backup mode is enabled.</td>
<td>• If a binary log file reaches 500 MB in size or its write operation exceeds 6 hours, the system starts to write data into a new binary log file. The old binary log file is then asynchronously uploaded to Alibaba Cloud Object Storage Service (OSS).</td>
</tr>
<tr>
<td>• MySQL 5.7 and MySQL 8.0 with standard or enhanced SSDs (in the High-availability Edition):</td>
<td>• After a binary log file is uploaded to OSS, it no longer occupies disk space on your RDS instance, but you can still use it for data restoration.</td>
</tr>
<tr>
<td>Support snapshot backup. Snapshot backups can be used to restore data to a new RDS instance. You cannot download these backups.</td>
<td>• The Basic Edition does not support the upload of binary log files to OSS.</td>
</tr>
<tr>
<td>• MySQL 5.7 and MySQL 8.0 with standard SSDs (in the Basic Edition):</td>
<td>• You cannot access the OSS buckets that store binary log files.</td>
</tr>
<tr>
<td>Support snapshot backup. Snapshot backups can be used to restore data to a new RDS instance. You cannot download these backups.</td>
<td></td>
</tr>
</tbody>
</table>

### Configure a backup policy that is used to automatically back up your RDS instance

ApsaraDB for RDS automatically backs up your RDS instance based on the specified backup policy.

1. Log on to the [ApsaraDB for RDS console](#).
2. In the top navigation bar, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID to open the Basic Information page.

4. In the left-side navigation pane, click Backup and Restoration.

5. On the Backup and Restoration page, click the Backup Settings tab and then the Edit button.

6. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Cycle</td>
<td>The cycle to create backups. You can select one or more days of a week.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> To ensure data security, we recommend that you back up your RDS</td>
</tr>
<tr>
<td></td>
<td>instance at least twice a week.</td>
</tr>
<tr>
<td>Backup Time</td>
<td>The period of time during which you want to create a backup. We recommend</td>
</tr>
<tr>
<td></td>
<td>that you select an off-peak hour.</td>
</tr>
</tbody>
</table>
### Retention Period

The period of time for which you want to retain backups. You can specify a specific number of days or select **long term backup**.

- If you specify a specific number of days, the system deletes backups whose retention reaches the specified period. If you select the **long term backup** option, the system does not delete backups. If you want to retain backups after you release the RDS instance, set the **Backup Retention Policy After Release** parameter to **Latest** or **All**.
- If the specified retention period does not exceed 730 days, backups are retained as regular backups.
- If the specified retention period exceeds 730 days, backups whose retention exceeds 730 days are automatically converted into archived backups. Therefore, you must also specify an archived backup retention period.

### Log Backup

The switch to enable or disable the log backup function.

**Notice:**
If you disable this function, all binary log files are deleted, and you cannot restore the RDS instance to a point in time.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Note:</th>
<th>Note:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log Retention Period</td>
<td>• The period of time for which you want to retain binary log files. Valid values: 7 to 730. Unit: days. Default value: 7.</td>
<td>If the RDS instance runs MySQL 5.7 and the Basic Edition based on standard SSDs, the log retention period is fixed to seven days and cannot be changed.</td>
<td>This parameter is available only when the RDS instance is used with local SSDs.</td>
</tr>
<tr>
<td></td>
<td>• The log retention period must be shorter than or equal to the data retention period.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Restore Individual</td>
<td>This function is used to restore individual databases and tables. After you enable this function, the system changes the backup file format to support this function. By default, this function is enabled and cannot be disabled. For more information, see Restore individual databases and tables of an ApsaraDB RDS for MySQL instance.</td>
<td>Note:</td>
<td>This parameter is available only when the RDS instance is used with local SSDs.</td>
</tr>
<tr>
<td>Database/Table</td>
<td></td>
<td>Note:</td>
<td></td>
</tr>
<tr>
<td>Backup Retention Policy</td>
<td>The policy that is used to retain data backup files after the RDS instance is released. Valid values: None, Latest, and All.</td>
<td>This parameter is available only when the RDS instance is used with local SSDs.</td>
<td></td>
</tr>
<tr>
<td>After Release</td>
<td>To prevent data losses caused by overdue payments or unintended operations, we recommend that you select Latest or All.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Manually back up your RDS instance**

In this example, your RDS instance runs MySQL 5.7 and the High-availability Edition based on local SSDs.

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID to open the Basic Information page.

4. In the upper-right corner of the page, click **Back Up Instance**.

5. Select a backup mode and click **OK**.

**Note:**
If you select the *Logical Backup* mode and then the *Single-Database Backup* policy, you must also select the databases that you want to back up from the left-side list and click > to add them to the right-side list. If no databases are available, you must create
databases. For more information, see Create databases and accounts for an ApsaraDB RDS MySQL instance.

6. In the upper-right corner of the page, click the Task Progress button to view the progress of the backup task.

Note:
- After the backup task is complete, you can go to the Backup and Restoration page to download the data backup file. Some RDS instances do not support the download of
backup files. For more information, see Download data backup files and binary log files.

- The retention period of manually created backups is determined by the Retention Period parameter. For more information, see the parameters described in step 6.

FAQ

1. Can I disable the data backup function of my RDS instance?
   
   No, you cannot disable the data backup function of your RDS instance. However, you can reduce the backup frequency to as low as twice a week. The data backup retention period must span at least seven days.

2. Can I disable the log backup function of my RDS instance?
   
   Yes, you can disable the log backup function of your RDS instance if you are not using the Basic Edition.

3. Why did my backup task fail?
   
   You may have executed DDL statements when the backup task was being executed. DDL statements trigger locks on tables. Your backup task may have failed as a result of the table locks.

4. Why does my RDS instance have a small volume of data but the size of the generated snapshot backup file is large?
   
   When creating a single snapshot backup, the system eliminates empty blocks. This allows the size of the snapshot backup file to be smaller than the available disk capacity. Each block is 2 MB in size. If write operations are dispersed, a large number of blocks are not full. For example, 3 MB of data may be written into two, three, or four blocks, and none of these blocks is full. When calculating the size of the snapshot backup file, the system considers all of these non-empty blocks to which data is written. As a result, the storage space occupied by the snapshot backup file is greater than the actual size of the snapshot backup file.
5. After I release my RDS instance, how do I restore data by using a retained backup file?

Log on to the ApsaraDB for RDS console and navigate to the **Backup for Deleted Instances** page. Then, download the backup file that you need and restore data to an on-premises database.

References

- Download data backup files and binary log files
- Restore the data of an RDS MySQL instance

Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_241</td>
<td>Creates a backup for an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_242</td>
<td>Queries the backups created for an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_243</td>
<td>Queries the backup settings of an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_244</td>
<td>Modifies the backup settings of an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_245</td>
<td>Deletes the data backup files of an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_246</td>
<td>Queries the backup tasks created for an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_247</td>
<td>Queries the binary log files of an ApsaraDB for RDS instance.</td>
</tr>
</tbody>
</table>
13.2 View the quota of free backup space for an ApsaraDB RDS for MySQL instance

This topic describes how to view the quota of free backup space for an ApsaraDB RDS for MySQL instance and how to calculate the backup space usage beyond the quota. The quota varies based on the database engine version and RDS edition used.

Backup files occupy backup space. Each RDS instance is allocated a quota of free backup space. If the total size of backup files exceeds the quota, additional fees are incurred.

**Calculate the quota of free backup space and the backup space usage beyond the quota**

RDS for MySQL instances equipped with standard or enhanced SSDs support only snapshot backups. The quota that is allocated to store snapshot backups is calculated based on the following formula: Quota = 200% × Storage capacity purchased during instance creation (unit: GB; rounded up).

RDS for MySQL instances equipped with local SSDs support both physical and logical backups. The quota that is allocated to store physical and logical backups is calculated based on the following formula: Quota = 50% × Storage capacity purchased during instance creation (unit: GB; rounded up).

The backup space usage beyond the quota is calculated based on the following formula:

Backup space usage beyond the quota = Data backup volume + Log backup volume - Quota

For example, if the data backup volume is 30 GB, the log backup volume is 10 GB, and the storage capacity purchased during instance creation is 60 GB, then the hourly rate of backup space usage beyond the quota is calculated as follows: Hourly rate of backup space usage beyond the quota = 30 + 10 - 50% × 60 = 10 (GB). This means that you must pay for an additional 10 GB of storage space per hour.

**Note:**

- For more information about the hourly rate of backup space usage beyond the quota, see the pricing information at ApsaraDB for RDS.
• The Basic Edition used with some database engines allows you to store backups that are generated during the last seven days. The storage is free of charge. For more information, go to the ApsaraDB for RDS console.

View the quota of free backup space in the ApsaraDB for RDS console

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID to open the Basic Information page.
4. In the Usage Statistics section at the lower part of the page, view the quota of free backup space to the right of Disk Size.

Note:
The quota of free backup space varies based on the database engine version and RDS edition. The following figure only shows an example.

FAQ
• Will backup files occupy the storage space that I purchased during instance creation?
  No, backup files do not occupy the storage space that you purchased during instance creation.
• Can I purchase subscription-billed backup space?
  No, you cannot purchase backup space.
13.3 Download data backup files and binary log files

This topic describes how to download unencrypted data backup files and binary log files from an ApsaraDB RDS for MySQL instance. You can archive the files for future use or use them to restore data to an on-premises database.

Limits

A RAM user with only read-only permissions cannot download files. You can grant the required permissions to the RAM user in the RAM console. For more information, see Grant backup file download permissions to a RAM user with only read-only permissions.

<table>
<thead>
<tr>
<th>Database engine</th>
<th>Download of data backup files</th>
<th>Download of binary log files</th>
</tr>
</thead>
</table>
| MySQL           | • MySQL 5.5, MySQL 5.6, MySQL 5.7, and MySQL 8.0 with local SSDs (in the RDS High-availability or Enterprise Editions) support the download of full physical or logical data backup files.  
• MySQL 5.7 and MySQL 8.0 with standard SSDs or enhanced SSDs (ESSDs) (in the RDS Basic or High-availability Editions) do not support the download of snapshots. | All MySQL versions support the download of binary log files.  
Note: For information about how to view binary log files, see #unique_250 |

Procedure

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target instance is located.

3. Find the target RDS instance and click its ID. The **Basic Information** page appears.

4. In the left-side navigation pane, click **Backup and Restoration**.

5. Click the Data Backup or Log Backup tab.
   - To download data backup files, click the **Data Backup** tab.
   - To download binary log files, click the **Log Backup** tab.

6. Specify a time range to query the data backup files or binary log files created during that period.

7. Find the data backup file or binary log file you want to download, and click **Download** in the **Actions** column.

---

**Note:**

- If the Download button is not available, see the **Limits** section in this topic.
- If you want to restore data by using a data backup file, select the file that was created at the point in time closest to the point in time requiring data restoration.
- If you use a binary log file to restore data to an on-premises database, make sure that the following requirements are met:
  - The Instance ID of the binary log file on the Log Backup tab must be the same as the Instance No. displayed on the Data Backup tab.
  - The start time of the binary log file must be later than the start time of the specified time range and earlier than the point in time requiring data restoration.
8. In the Download Binary Log dialog box that appears, click Download.

We currently offer free downloads of binlog files for a limited period of time. If your ECS and RDS instances are in the same region, accessing an internal download URL to download binlog files increases the level of security and download speed.

**Note:** The latest version of Flash is required to copy the download URL.

<table>
<thead>
<tr>
<th>Download method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download</td>
<td>Use a browser to download the backup file.</td>
</tr>
<tr>
<td>Copy Internal Download URL</td>
<td>To only copy the internal download URL of the file. If your ECS instance and RDS instance reside in the same region, you can log on to the ECS console and use the internal download URL to download the file. This method is faster and more secure.</td>
</tr>
<tr>
<td>Copy External Download URL</td>
<td>To only copy the external download URL of the file. If you want to use other tools to download the file, select this method.</td>
</tr>
</tbody>
</table>

**Note:**
If you are using a Linux operating system, you can run the following command to save the file:

```
wget -c '"<The URL from which to download the file>"' -O "<The name of the file you want to download>
```

- The `-c` parameter enables resumable download.
- The `-O` parameter saves the downloaded file by its specified name. We recommend that you use the file name from the download URL.
- If the download URL contains more than one parameter, we recommend that you enclose the download URL in a pair of single quotation marks (`'`).
FAQ

1. Why do I find two binary log files with the same name on the Log Backup tab?

   In the High-availability Edition, your database system consists of a primary instance and a secondary instance. The primary and secondary instances both generate binary log files. Each binary log file is identified by an **Instance ID** on the Log Backup tab. You can distinguish the binary log files generated by the primary instance from those generated by the secondary instance based on their instance IDs. On the **Service Availability** page, you can view the primary instance ID under Primary Instance No. and the secondary instance ID under Secondary Instance No..

2. What can I do with the data backup file or binary log file that I downloaded?

   You can use a data backup file or binary log file to restore data at any time. For more information, see [Use a physical backup file to restore an ApsaraDB RDS for MySQL instance to a user-created MySQL database](#) or [Restore data from logical backup files of an ApsaraDB RDS MySQL instance to an on-premises database](#).

3. Why is the Download button not available on the Backup and Restoration page?

   For MySQL 5.7 and MySQL 8.0 instances with standard SSDs or enhanced SSDs (ESSDs) (in the RDS Basic or High-availability Editions), backup files cannot be downloaded. You can [restore data to a new RDS instance](#) or use DTS to migrate data to an on-premises database.
13.4 Back up an ApsaraDB RDS for MySQL instance across regions

This topic describes how to back up an ApsaraDB RDS for MySQL instance by using the cross-region backup function. This function automatically replicates the backup files of your RDS instance from the source region to a specified destination region. The backup files in the destination region are used to manage and restore your RDS instance.

Prerequisites

Your RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 8.0 High-availability Edition (with local SSDs)
- MySQL 5.7 High-availability Edition (with local SSDs)
- MySQL 5.6

Context

The cross-region backup function is different from the default backup function. For more information about the default backup function, see Back up an ApsaraDB RDS for MySQL instance.

You can use a cross-region backup file of your RDS instance to restore data to a new RDS instance or an existing instance in the destination region. For more information, see Restore an ApsaraDB RDS for MySQL instance across regions.

Differences between cross-region and default backup

- The cross-region backup function is disabled by default and must be enabled manually. The default backup function is enabled by default.
- Cross-region backup files are stored in the destination region. Default backup files are stored in the source region.
- Cross-region backup files can be restored to a new RDS instance or an existing instance in the source or destination region. Default backup files can be restored to the original RDS instance or a new RDS instance in the source region.
- Cross-region backup files are independent of your RDS instance. If you release your RDS instance, cross-region backup files will be retained based on the specified retention period, but default backup files will only be retained for seven days.

Billing

The fees for the cross-region backup function include the following two parts:
• Remote storage: USD 0.0002/GB/hour.
• Traffic consumption: For more information, see Billing methods and items.

Precautions

• Cross-region backup files can be restored to the source or destination region. However, if Transparent Data Encryption (TDE) is enabled, you can only restore them to the source region.
• Cross-region backups do not affect default backups. Both types of backups can be performed at the same time. Cross-region backups are used to replicate default backup files from the source to destination regions.
• After a default backup is complete, the system will trigger a cross-region backup to dump the generated default backup file to the destination region.
• If the cross-region data backup function is enabled and no valid data backup files are generated within 24 hours, the system will trigger a data backup on the secondary instance.
• If the cross-region log backup function is enabled, the system will check for valid data backup files generated within the last 24 hours.
  - If continuous binary logs are generated following a valid data backup file, the system will dump these binary logs.
  - If no continuous binary logs are generated following a valid data backup file, the system will trigger a data backup on the secondary instance.
• Cross-region backup is not supported in some regions. The following table lists the regions that support the cross-region backup function.

<table>
<thead>
<tr>
<th>Source region</th>
<th>Destination region</th>
</tr>
</thead>
<tbody>
<tr>
<td>China (Hangzhou)</td>
<td>China (Shanghai), China (Qingdao), China (Shenzhen)</td>
</tr>
<tr>
<td>China (Shanghai)</td>
<td>China (Hangzhou), China (Qingdao), China (Shenzhen)</td>
</tr>
<tr>
<td>China (Qingdao)</td>
<td>China (Hangzhou), China (Shanghai), China (Shenzhen)</td>
</tr>
<tr>
<td>China (Beijing)</td>
<td>China (Hangzhou), China (Shanghai), China (Qingdao), China (Shenzhen)</td>
</tr>
<tr>
<td>China (Shenzhen)</td>
<td>China (Hangzhou), China (Shanghai), China (Qingdao)</td>
</tr>
<tr>
<td>China (Hong Kong)</td>
<td>China (Hangzhou), China (Shanghai), China (Qingdao), China (Shenzhen)</td>
</tr>
</tbody>
</table>
Enable cross-region backup for an instance

1. Log on to the ApsaraDB for RDS console.

2. In the upper-left corner of the page, select the region where the instance is located.

3. Find the instance. In the Actions column corresponding to the instance, choose More > Cross-region Backup Settings.

Note:

- You can also click Backup and Restoration in the left-side navigation pane and then click Edit in the upper-left corner of the Cross-region Backup tab.
- If the Cross-region Backup tab is not displayed, make sure the instance meets the prerequisites.
4. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cross-region Backup Status</strong></td>
<td>Specifies whether to enable cross-region backup. Select <strong>Enable</strong>.</td>
</tr>
<tr>
<td><strong>Backup Region</strong></td>
<td>The destination region to which backup files are stored. The system replicates the backup files generated on the RDS instance to OSS buckets in this region.</td>
</tr>
<tr>
<td><strong>Cross-region Retention Period</strong></td>
<td>The retention period of the backup data in days. You can specify an integer from 7 to 1825. Cross-region backup files can be retained for a maximum of five years.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>After the RDS instance expires or is released, its cross-region backup files are still retained based on the specified retention period. You can log on to the ApsaraDB for RDS console and navigate to the Cross-region Backup tab of your RDS instance to view the retained cross-region backup files.</td>
</tr>
<tr>
<td><strong>Cross-region Log Backup Status</strong></td>
<td>Specifies whether to enable the cross-region log backup function. After you enable this function, the system replicates the binary log files generated on the RDS instance to OSS buckets in the specified destination region.</td>
</tr>
</tbody>
</table>
5. Click OK.

Enable cross-region backup for multiple instances

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the instance is located.
3. In the left-side navigation pane, click Cross-region Backup.
4. Select the Pending Instances tab.
5. Select instances for which you want to enable cross-region backup, and then click Backup Settings.

Note:
You can also click Settings corresponding to an instance to enable cross-region backup for a single instance.
6. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-region Backup Status</td>
<td>Specifies whether to enable cross-region backup. Select <strong>Enable</strong>.</td>
</tr>
<tr>
<td>Backup Region</td>
<td>The destination region to which backup files are stored. The system replicates the backup files generated on the RDS instance to OSS buckets in this region.</td>
</tr>
<tr>
<td>Cross-region Retention Period</td>
<td>The retention period of the backup data in days. You can specify an integer from 7 to 1825. Cross-region backup files can be retained for a maximum of five years.</td>
</tr>
<tr>
<td>Note:</td>
<td>After the RDS instance expires or is released, its cross-region backup files are still retained based on the specified retention period. You can log on to the ApsaraDB for RDS console and navigate to the Cross-region Backup tab of your RDS instance to view the retained cross-region backup files.</td>
</tr>
<tr>
<td>Cross-region Log Backup Status</td>
<td>Specifies whether to enable the cross-region log backup function. After you enable this function, the system replicates the binary log files generated on the RDS instance to OSS buckets in the specified destination region.</td>
</tr>
</tbody>
</table>

7. Click **OK**.
Modify cross-region backup settings

The cross-region backup menu is added to the RDS console. You can modify the cross-region backup settings even if the instance is released.

1. Log on to the ApsaraDB for RDS console.
2. In the left-side navigation pane, click Cross-region Backup.
3. Find the instance and click Settings in the Cross-region Backup Settings column to modify the backup settings.

Note:
If the instance is released, you can only modify the retention period.

Disable cross-region backup

To disable the cross-region backup function, follow these steps:

1. Log on to the ApsaraDB for RDS console.
2. In the left-side navigation pane, click Cross-region Backup.
3. Find the target RDS instance, and click Settings in the Cross-region Backup Settings column.
4. Set Cross-region Backup Status to Disabled and set Cross-region Retention Period to 7 days.

Note:
After you disable cross-region backup, no new backup files are generated, but the existing backup files are retained for at least seven days. You must set the cross-region retention period to seven days. After the retention period elapses, all existing backups are deleted and you are no longer charged for them.

5. Click OK.

View the cross-region backup settings

1. Log on to the ApsaraDB for RDS console.
2. In the left-side navigation pane, click **Cross-region Backup**. Then, you can view the cross-region backup settings of all RDS instances.

---

**FAQ**

**Why am I still incurring charges after I disable the cross-region backup function?**

After you disable cross-region backup, it no longer consumes traffic, and your RDS instance no longer generates new backup files. However, you must still pay for the storage space that is occupied by the existing backup files during the specified retention period. The existing backup files are retained for at least seven days. You can set the retention period to seven days. After the retention period elapses, all existing backups are deleted, and you are no longer charged for them.

**Related operations**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_254</td>
<td>Checks whether a cross-region backup file of an ApsaraDB for RDS instance is used for cross-region restoration.</td>
</tr>
<tr>
<td>#unique_255</td>
<td>Restores the data of an ApsaraDB for RDS instance to a new instance in a region different from the original instance.</td>
</tr>
<tr>
<td>#unique_256</td>
<td>Modifies the cross-region backup settings of an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_257</td>
<td>Queries the cross-region backup settings of an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_258</td>
<td>Queries the cross-region data backup files of an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_259</td>
<td>Queries the cross-region binary log files of an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_260</td>
<td>Queries the available destination regions that support cross-region backup from a region.</td>
</tr>
<tr>
<td>Operation</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>#unique_261</td>
<td>Queries the time period to which a data backup file can be restored.</td>
</tr>
<tr>
<td>#unique_262</td>
<td>Queries the cross-region backup settings of ApsaraDB for RDS instances in a region.</td>
</tr>
</tbody>
</table>
14 Restoration

14.1 Overview of data recovery

This topic provides an overview of the data recovery solutions supported by ApsaraDB RDS for MySQL.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Reference</th>
</tr>
</thead>
</table>
| Restore the data of an ApsaraDB RDS for MySQL database to its original instance or to a new instance | • Restore the data of an RDS MySQL instance  
• Restore individual databases and tables of an ApsaraDB RDS for MySQL instance  
• Restore an ApsaraDB RDS for MySQL instance across regions |
| Restore the data of an ApsaraDB RDS for MySQL database to a user-created database | • Use a physical backup file to restore an ApsaraDB RDS for MySQL instance to a user-created MySQL database  
• Restore data from logical backup files of an ApsaraDB RDS MySQL instance to an on-premises database |

Note:
If the original instance is not released or you do not want to restore data from a specific point in time, we recommend that you use DTS to migrate the data from your ApsaraDB for MySQL database to a user-created MySQL database.

14.2 Restore the data of an RDS MySQL instance

This topic describes how to restore the data of an RDS MySQL instance by using a data backup.

For more information about how to restore the data in other database engines, see the following topics:

• #unique_265  
• #unique_266  
• #unique_267
Background information

You can use any of the following methods to restore the data of an RDS for MySQL instance:

- Method 1: restore data to a new RDS instance, verify the data, and then migrate the data back to the original RDS instance. This article introduces this method.
- Method 2: restore single-database or single-table data to the original RDS instance or a new RDS instance. For more information, see Restore individual databases and tables of an ApsaraDB RDS for MySQL instance.
- Method 3: restore data to a new instance or an existing instance in the destination region. For more information, see Restore an ApsaraDB RDS for MySQL instance across regions.

Note:

For more information about how to restore data to a user-created database, see Use a physical backup file to restore an ApsaraDB RDS for MySQL instance to a user-created MySQL database or Restore data from logical backup files of an ApsaraDB RDS MySQL instance to an on-premises database.

Prerequisites

The original instance must meet the following conditions:

- It is running and is not locked.
- It does not have an ongoing migration task.
- To restore data by time, you must make sure that the log backup function is enabled.
- To restore data by backup set, you must make sure that the original instance has at least one backup set.

Precautions

- The new instance must have the same whitelist, backup, and parameter settings as the original instance.
- The data information of the new RDS instance must be the same as that of the created RDS instance or from the specified time point.
- The new RDS instance carries the account information in the used backup file or that from the specified time point.
Billing

The data is restored to the new instance, so the new instance fee is charged. For more information, see #unique_78.

Note:
If you use DTS to migrate data from a new instance to the original instance, you are not charged for schema migration and full migration.

Restore data to a new instance

1. Log on to the ApsaraDB for RDS console.
2. Select the region where the target instance is located.
3. Click the ID of the instance.
4. In the left-side navigation pane, select Backup and Restoration.
5. In the upper-right corner of the page, click Restore Database (Previously Clone Database).
6. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Billing Method    | • **Subscription**: You must pay the subscription fee when you create an instance. We recommend that you select subscription billing for long-term use because it is more cost-effective than pay-as-you-go billing. You will receive larger discounts for longer subscription periods.  
• **Pay-As-You-Go**: A pay-as-you-go instance is charged hourly based on your actual resource usage. We recommend that you select pay-as-you-go billing for short-term use. You can release your pay-as-you-go instance to reduce costs when you no longer need it. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Restore Mode</strong></td>
<td></td>
</tr>
</tbody>
</table>
| - **By Time**: allows you to restore data to a point in time within the specified log retention period. The time is accurate to seconds. For more information about how to view or change the retention period of log backups, see Back up an ApsaraDB RDS for MySQL instance.  
| - **By Backup Set**: allows you to restore data from a data backup file.  
| **Note:** By Time is displayed only when the log backup feature is enabled. |
| **Zone of Primary Node** | The zone of primary node where the RDS instance resides. Each zone is an independent physical location within a region. There is no substantive difference between zones in the same region. If your database system spans multiple zones, it provides zone-level disaster recovery.  
You only need to select a primary zone. The system automatically assigns a secondary zone to the RDS instance. |
| **Deployment Method** | You can set the instance as multi-zone or single-zone deployment. |
| **Instance Type** |  
| - **Entry-level**: belongs to the general-purpose instance family. A general-purpose instance occupies the exclusive memory and I/O resources allocated to it, but shares CPU and storage resources with the other general-purpose instances that are deployed on the same server.  
| - **Enterprise-level**: belongs to the dedicated instance family. A dedicated instance occupies exclusive CPU, memory, storage, and I/O resources. The top configuration of the dedicated instance family is the dedicated host. A dedicated host instance occupies all CPU, memory, storage, and I/O resources on the server where it resides.  
| **Note:** Each instance type supports a specific number of CPU cores, memory capacity, maximum number of connections, and maximum input/output operations per second (IOPS). For more information, see unique_82.
### Parameter | Description
--- | ---
**Storage** | The storage space of the instance, including the space for data, system files, binary log files, and transaction files. The storage capacity increases in increments of 5 GB.

**Note:** The dedicated instance family supports exclusive allocations of resources. Therefore, the storage capacity of each instance type with local SSDs in this family is fixed. For more information, see #unique_82.

7. Click **Next: Instance Configuration**.

8. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Network Type** | • **Classic Network**: a traditional type of network.  
• **VPC**: A virtual private cloud (VPC) is an isolated virtual network with higher security and better performance than the classic network. If you select the VPC network type, you must also specify **VPC** and **VSwitch of Primary Node**.  

**Note:** Make sure that the RDS instance has the same network type as the ECS instance that you want to connect. If the RDS and ECS instances have different network types, they cannot communicate over an internal network. If the network types of the RDS and ECS instances are both VPC, you must also make sure that they reside in the same VPC. |

9. Click **Next: Confirm Order**.

10. Confirm the settings in the **Parameters** section, specify **Purchase Plan** and **Duration**, read and select Terms of Service, and then click **Pay Now**. You need to specify Duration only when you create a subscription instance.

**Note:**
For subscription instances, we recommend that you select **Auto-Renew Enabled**, which will free you from regular manual renewals and avoid business interruptions.

Log on to the new instance and verify the data

For more information about how to log on to an instance, see [connect to an instance](#).

Migrate data to the original instance

After you verify the data in the new instance, you can migrate the data that you require from the new instance to the original instance. For more information, see [Migrate data between ApsaraDB for RDS instances](#).

**Note:**

Data Migration refers to migrating data from one RDS instance (the source RDS instance) to another (the destination RDS instance). The data migration operation does not interrupt the source RDS instance.

**FAQ**

- **How can I restore a database that was accidentally deleted?**

  You can restore a single database. For more information, see [Restore individual databases and tables of an ApsaraDB RDS for MySQL instance](#). For instances that do not support data restoration, see this topic to restore all data to a new instance. After verification, migrate the data back to the original instance.
Can I restore data by time point without data backup?

No. To restore data to a specified time point, you must first restore full backup data that was generated before the selected time point to the instance. Then, you can restore incremental data to the selected time point based on binlogs.

Why can't I select a vSwitch on the primary node during restoration?

Because no vSwitch is available in the zone you selected in the previous step (basic configuration), you cannot select a primary vSwitch in the current step (network and resource group). You can click create in the console in the VPC console, create a vSwitch in the zone to select the primary vSwitch.

14.3 Restore individual databases and tables of an ApsaraDB RDS for MySQL instance

This topic describes how to restore the individual databases and tables that are unintendedly deleted from an ApsaraDB RDS for MySQL instance. You can quickly restore them by using backups.

For more information about data restoration at the instance level, see Restore the data of an RDS MySQL instance.

Prerequisites

- The RDS instance is running one of the following MySQL versions and RDS editions:
  - MySQL 8.0 in the High-availability Edition (with local SSDs)
  - MySQL 5.7 in the High-availability Edition (with local SSDs)
  - MySQL 5.6 in the High-availability Edition
- Instance storage engine is not X-Engine.
- The number of tables on the RDS instance does not exceed 50,000.
The Restore Individual Database/Table function is enabled. To enable this function, follow these steps: Log on to the ApsaraDB for RDS console, navigate to the **Backup and Restoration** page, click the **Backup Settings** tab and then the Edit button on the tab, and in the dialog box that appears select Enabled for Restore Individual Database/Table.

**Note:**
- After this function is enabled, a new backup format will be used to support the restoration of individual databases and tables. This function cannot be disabled after it is enabled.
- This function is enabled for each new RDS instance by default and cannot be disabled.

If you want to restore individual databases and tables to the original RDS instance, the original RDS instance must meet the following requirements:

- The original RDS instance is in the Running state and is not locked.
- The original RDS instance does not have an ongoing migration task.
- If you want to restore individual databases and tables to a point in time, the log backup function is enabled for the original RDS instance.
- If you want to restore individual databases and tables from a backup set, the original RDS instance has at least one backup set.
- If you want to restore individual databases and tables to a new RDS instance, the original RDS instance must meet the following requirements:
  - The original RDS instance is in the Running state and is not locked.
  - If you want to restore individual databases and tables to a point in time, the log backup function is enabled for the original RDS instance.
  - If you want to restore individual databases and tables from a backup set, the original RDS instance has at least one backup set.

**Precautions**

- When you restore individual databases and tables to the original RDS instance, a primary/secondary switchover is triggered. This may incur a brief disconnection. Make sure that your application is configured to automatically reconnect to the original RDS instance.
- The Restore Individual Database/Table function triggers a format conversion of backup files from the TAR format to the xbstream format. As a result, the storage space occupied by these backup files slightly increases. Keep an eye on your backup space usage. If your backup space usage exceeds the capacity of the specified free tier, additional fees are incurred. We recommend that you specify a proper backup cycle that best meets your business needs and maximizes the usage of backup space within the free tier.
- The Restore Individual Database/Table function only restores the specified tables. You must select all of the tables that you want to restore. The restoration will fail in the following scenarios:
  - The tables that you specified are deleted during the period spanning from the point in time when the last backup set was generated to the point in time to which you want to restore data.
  - The restoration involves a table that you have not specified. For example, you specified to restore Table B, but Table B is renamed from Table A before the specified point in time. In this situation, the restoration fails because you did not specify Table A.

If you cannot identify all of the tables involved, you can only restore data at the instance level. For more information, see Restore the data of an RDS MySQL instance.

- You can select a maximum of 50 databases or tables at a time.

**Procedure**

1. Log on to the ApsaraDB for RDS console.
2. Select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID.

4. In the left-side navigation pane, click **Backup and Restoration**.

5. In the upper-right corner of the page, click **Restore Individual Database/Table**.

**Note:**
If the **Restore Individual Database/Table** button does not appear, see the "Prerequisites" section.

6. In the dialog box that appears, configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Restore To** | • **Current Instance**: restores individual databases and tables to the original RDS instance.  
• **New Instance**: restores individual databases and tables to a new RDS instance. |
| **Restore Method** | • **By Backup Set**  
• **By Time**: If you select this option, you can specify a point in time within the specified log retention period and restore data to that point in time. For more information about viewing and modifying the log retention period, see [Back up an ApsaraDB RDS for MySQL instance](#).  
**Note:**  
The **By Time** option appears only when the log backup function is enabled. |
| **Backup Set** | Select the backup set from which you want to restore individual databases and tables.  
**Note:**  
This parameter appears only when you set the **Restore Method** parameter to **By Backup Set**. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Restore Time</strong></td>
<td>Select the point in time to which you want to restore individual databases and tables.</td>
</tr>
<tr>
<td></td>
<td><img src="https://example.com" alt="Note:" /> This parameter appears only when you set the <strong>Restore Method</strong> parameter to <strong>By Time</strong>.</td>
</tr>
<tr>
<td><strong>Restore Mode</strong></td>
<td>• <strong>Logical Restoration</strong>: The restoration is slow.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Physical Restoration</strong>: The restoration is fast, but incurs a primary/ secondary switchover. In addition, all of the read-only instances attached with the original RDS instance will be restarted. If the original RDS instance is being maintained, the volume of data that you want to restore is small, or the data replication with the read-only instances is interrupted, the backend automatically selects <strong>Logical Restoration</strong>.</td>
</tr>
<tr>
<td></td>
<td><img src="https://example.com" alt="Note:" /> This parameter appears only when the original RDS instance has read-only instances.</td>
</tr>
<tr>
<td><strong>Databases and Tables to Restore</strong></td>
<td>Select the databases and tables that you want to restore.</td>
</tr>
</tbody>
</table>
ApsaraDB for RDS

RDS MySQL Database / 14 Restoration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Selected Databases and Tables | • This section displays the selected databases and tables. You can specify new names for these databases and tables.  
• This section also displays the total size of the selected databases and tables and the remaining storage space. Make sure that the remaining storage space is sufficient before the restoration. |

### Restore Individual Database/Table

- **Restore To**: ![Current Instance](on) ![New Instance](off)
- **Restore Method**: ![By Backup Set](off) ![By Time](on)
- **Restore Time**: Sep 6, 2019 16:53

**Databases and Tables to Restore**

- **Search by database name**: ![Search by database name](on)
- **Search by table name**: ![Search by table name](on)

  - **Could not find any record that met the condition.**
  - **Could not find any record that met the condition.**

**Selected Databases and Tables**

<table>
<thead>
<tr>
<th>Database Name</th>
<th>New Database Name</th>
<th>Table Name</th>
<th>New Table Name</th>
<th>Actions</th>
</tr>
</thead>
</table>

**Total Storage of Selected**: 0M
**Available Storage of Current Instance**: 29487M

OK  Cancel
7. Click **OK**.

**Note:**
If you set the **Restore To** parameter to **New Instance**, you are directed to the page where you can purchase a new RDS instance. On this page, you need to configure the following parameters and complete the payment.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Edition** | • Basic Edition: The database system consists of only one instance. Computing is separated from storage. This edition is cost-effective, but is not recommended for production environments.  
• High-availability Edition: The database system consists of one primary instance and one secondary instance. These instances work in the classic high-availability architecture.  
• Enterprise Edition: The database system consists of one primary instance and two secondary instances. These instances reside in three different zones of the same region to provide finance-level reliability. The Enterprise Edition is available only in four regions: China (Hangzhou), China (Shanghai), China (Shenzhen), and China (Beijing).  
For more information, see [ApsaraDB for RDS edition overview](#).  
The editions available vary based on the database engine version that you select. |
| **Zone** | The zone where the new RDS instance resides. Each zone is an independent physical location within a region. Zones in the same region provide the same services.  
You can create the new RDS instance and the ECS instance to which you want to connect in the same zone or in different zones.  
**Note:** The new RDS instance must reside in the same region as the original RDS instance. You cannot change the region. |
| **Instance Type** | Each instance type supports a specific number of CPU cores, memory capacity, maximum number of connections, and maximum input/output operations per second (IOPS). For more information, see [Primary instance types](#). |
| **Capacity** | The storage capacity that the new RDS instance has available to store data files, system files, binary log files, and transaction files. |
### Network Type

- **Classic Network**: a traditional type of network.
- **VPC**: This is the recommended network type. A Virtual Private Cloud (VPC) is an isolated virtual network that provides higher security and better performance than a classic network.

### FAQ

- After the format of a backup file is converted from TAR to xbstream, will it become unavailable?

  No, the backup file is still available after a format conversion from TAR to xbstream.

- Why does the Restore Individual Database/Table function suddenly become unavailable?

  Check whether the number of tables on your RDS instance exceeds 50,000. If the number exceeds 50,000, the Restore Individual Database/Table function is unavailable.

### Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_271</td>
<td>Restores individual databases and tables of an ApsaraDB for RDS instance to the same instance.</td>
</tr>
<tr>
<td>#unique_272</td>
<td>Restores individual databases and tables of an ApsaraDB for RDS instance to a new instance.</td>
</tr>
</tbody>
</table>

### 14.4 Restore an ApsaraDB RDS for MySQL instance across regions

If you have backed up an instance across regions, you can restore data from the backup file to a new instance or an existing instance in the region where the source instance is located or the region where the cross-region backup file is stored.

### Prerequisites

*An ApsaraDB RDS for MySQL instance has been backed up across regions.*
Precautions

If the source instance has enabled the database proxy feature and does not have a privileged account, you must reset the password to connect to the instance to which data is restored.

Restore cross-region backup data to a new instance

1. Log on to the ApsaraDB for RDS console.
2. In the left-side navigation pane, click Cross-region Backup.
3. Find the destination RDS instance and click its ID.
4. On the Database Backup tab, find the target backup set and click Restore in the corresponding Actions column.
5. Select Restore to New Instance and click OK.
6. On the Restore Database page, select Subscription or Pay-As-You-Go and configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore Mode</td>
<td>• By Backup Set: Restore the data of the backup set to a new instance.</td>
</tr>
<tr>
<td></td>
<td>• By Time: Restore data from a specified time point to a new instance. You</td>
</tr>
<tr>
<td></td>
<td>can specify any time point within the log backup retention period.</td>
</tr>
<tr>
<td>Backup Set</td>
<td>When Restore Mode is set to By Backup Set, select the backup set that you</td>
</tr>
<tr>
<td></td>
<td>want to restore.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Restore Point</td>
<td>When <strong>Restore Mode</strong> is set to <strong>By Time</strong>, select the time point to restore data.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Local and cross-region logs can be restored to a specified time point.</td>
</tr>
<tr>
<td>Region</td>
<td>The region where the new instance is located.</td>
</tr>
<tr>
<td>Zone</td>
<td>An independent physical area within a region. There are no substantive differences between zones. You can choose to create RDS and ECS instances in the same zone or in different zones.</td>
</tr>
<tr>
<td>CPU and Memory</td>
<td>The type of the RDS instance. Each instance type supports a specific number of CPU cores, memory capacity, maximum number of connections, and maximum IOPS. For more information, see #unique_82.</td>
</tr>
<tr>
<td>Capacity</td>
<td>The storage capacity used to store data files, system files, binary log files, and transaction files.</td>
</tr>
<tr>
<td>Network Type</td>
<td>• <strong>Classic Network</strong>: a traditional type of network.</td>
</tr>
<tr>
<td></td>
<td>• <strong>VPC</strong>: the recommended network type. A VPC is an isolated virtual network with higher security and better performance than a classic network. If you select VPC, you must also select a VSwitch.</td>
</tr>
</tbody>
</table>

**Note:**
The database engine and edition are consistent with those of the source instance and cannot be modified.

7. Specify **Duration** (only applicable to subscription instances) and **Quantity**, and click **Buy Now**.

8. On the **Order Confirmation** page, select the check box to agree the terms of service and complete the payment as prompted.

**Restore cross-region backup data to an existing instance**

1. Log on to the ApsaraDB for RDS console.
2. In the left-side navigation pane, click **Cross-region Backup**.
3. Find the destination RDS instance and click its ID.

4. On the Database Backup tab, find the target backup set and click Restore in the corresponding Actions column.

5. Select Restore to Existing Instance and click OK.
6. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore Method</td>
<td>• <strong>By Backup Set</strong>: Restore the data of the backup set to an existing instance.</td>
</tr>
<tr>
<td></td>
<td>• <strong>By Time</strong>: Restore data from a specified time point to an existing instance. You can specify any time point within the log backup retention period.</td>
</tr>
</tbody>
</table>
### Parameter | Description
---|---
Region | The region where the destination instance is located.
Destination Instance | The destination instance to which the data is restored.
Databases and Tables to Restore | Select the databases and tables that you want to restore.
Selected Databases and Tables | Specify the database and table names after the databases and tables are restored to the destination instance. If you do not specify a name, the name before migration will be added with a _backup suffix to form a new name.

7. Click **OK**.

### References

After an RDS instance is created, you must configure whitelists and create accounts for it. If you want to connect to the RDS instance over the Internet, you must also apply for a public endpoint. After you complete these operations, you can connect to the RDS instance.

### Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_254</td>
<td>Checks whether the RDS instance has a cross-region backup set to perform the restoration.</td>
</tr>
<tr>
<td>#unique_255</td>
<td>Restores cross-region backup data to a new instance.</td>
</tr>
<tr>
<td>#unique_256</td>
<td>Modifies the cross-region backup settings of an RDS instance.</td>
</tr>
<tr>
<td>#unique_257</td>
<td>Queries the cross-region backup settings.</td>
</tr>
<tr>
<td>#unique_258</td>
<td>Queries the cross-region backup files of the data.</td>
</tr>
<tr>
<td>#unique_259</td>
<td>Queries the cross-region backup files of the logs.</td>
</tr>
<tr>
<td>#unique_260</td>
<td>Queries the available regions that support cross-region backup from the current region.</td>
</tr>
<tr>
<td>#unique_261</td>
<td>Queries the time period that contains the backup data that can be restored.</td>
</tr>
</tbody>
</table>
14.5 Use a physical backup file to restore an ApsaraDB RDS for MySQL instance to a user-created MySQL database

This topic describes how to use open source Percona XtraBackup to restore a physical backup file of an ApsaraDB RDS for MySQL instance to a user-created MySQL database.

Note:

- For information about how to back up an ApsaraDB RDS for MySQL instance, see #unique_273.
- Percona XtraBackup is not supported in Windows operating systems. For information about how to back up and restore an ApsaraDB RDS for MySQL instance that runs in a Windows operating system, see #unique_274.

Precautions

For example, your RDS instance runs MySQL 5.7 in a Linux 7 operating system.

- Percona XtraBackup is installed in the operating system. You can download the Percona XtraBackup software package from the official Percona XtraBackup website.
  - MySQL 5.6 and earlier versions require that you install Percona XtraBackup 2.3. For more information, see Percona XtraBackup 2.3.
  - MySQL 5.7 requires that you install Percona XtraBackup 2.4. For more information, see Percona XtraBackup 2.4.
  - MySQL 8.0 requires that you install Percona XtraBackup 8.0. For more information, see Percona XtraBackup 8.0.
- If your RDS instance runs MySQL 5.6 and is created after February 20, 2019, your data backup files are saved as xbstream compressed packages with the _qp.xb extension after they are downloaded.
- The user-created MySQL database runs in a 64-bit Linux operating system and uses the same MySQL version as your RDS instance.

Note:
You can only restore a data backup file of your ApsaraDB RDS for MySQL instance to a user-created MySQL database that runs in a Linux operating system.

Prerequisites

Your RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 8.0 in the High-availability Edition (with local SSDs)
- MySQL 5.7 in the High-availability Edition (with local SSDs)
- MySQL 5.6
- MySQL 5.5

Note:

Instances in the Basic Edition only provide snapshot backups, which cannot be downloaded. For more information, see the FAQ section.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click Backup and Restoration.
5. Click the Data Backup tab.
6. Specify a time range and click OK.
7. In the data backup file list, find the target data backup file, and click Download in the Actions column.

Note:

If the Download button does not appear, make sure that your MySQL version and RDS edition support the download of physical backup files. For more information, see Download data and log backup files.
8. In the **Download Instance Backup Set** dialog box that appears, click 

![Download Instance Backup Set](image1)

9. Log on to your ECS instance.

10. Run the following command to download the data backup file:

```
wget -c '<The external download URL of the data backup file>' -O <The name to use for the downloaded data backup file>
```

**Note:**

- The `-c` parameter enables resumable download.
- The `-O` parameter saves the downloaded data backup file based on its specified name. The specified file name must use the `.tar.gz`, `.xb.gz`, or `_qp.xb` extension that is included in the download URL.

11. Decompress the compressed package that you downloaded.

**Note:**

- You can replace the example custom path `/home/mysql/data` with the save path that you use for real scenarios.
- To decompress data backup files that are created by using the `innobackupex` tool, you must install `qpress`, which is a file archiver that you can download from the official [QuickLZ website](https://quicklz.com). After you download the `qpress` software package, run the following commands to perform an installation:

```
tar xvf qpress-1.11-linux-x64.tar
chmod 775 qpress
```
Physical backup files are downloaded in one of the following three formats:

- tar compressed package with the .tar.gz extension
- xbstream compressed package with the .xb.gz extension
- xbstream compressed package with the _qp.xb extension

**Note:**
If your RDS instance runs MySQL 5.6 and is created after February 20, 2019, your data backup files are saved as xbstream compressed packages with the _qp.xb extension after they are downloaded.

To decompress a tar compressed package with the .tar.gz extension, run the following command:

```
tar -izxvf <The name of the tar compressed package>.tar.gz -C /home/mysql/data
```

To decompress an xbstream compressed package with the .xb.gz extension, run the following command:

```
gzip -d -c <The name of the xbstream compressed package>.xb.gz | xbstream -x -v -C /home/mysql/data
```

To decompress an xbstream compressed package with the _qp.xb extension, run the following commands:

```
## Unpack the xbstream compressed package. cat <The name of the xbstream compressed package>_qp.xb | xbstream -x -v -C /home/mysql/data

## Decompress the xbstream compressed package when your RDS instance runs MySQL 5.6 or 5.7.
innobackupex --decompress --remove-original /home/mysql/data
## Decompress the xbstream compressed package when your RDS instance runs MySQL 8.0.
xtrabackup --decompress --remove-original --target-dir=/home/mysql/data
```

**Note:**
The -C parameter specifies the directory to which the xbstream compressed package is decompressed. This parameter is optional. If you do not specify it, the file is decompressed to the current directory.
12. Run the following command to query the files that are generated from the decompression:

```
ls -l /home/mysql/data
```

After the command is executed, the system displays information similar to the following. The blue part indicates the databases that are available on your RDS instance at the time when the data backup file was generated.

![ls command output]

13. Run the following commands to restore the data backup file to the user-created MySQL database:

```
## MySQL 5.6 or 5.7
innobackupex --defaults-file=/home/mysql/data/backup-my.cnf --apply-log /home/mysql/data

## MySQL 8.0
xtrabackup --prepare --target-dir=/home/mysql/data
xtrabackup --datadir=/var/lib/mysql --copy-back --target-dir=/home/mysql/data
```

If the system displays information similar to the following, the data backup file is restored to the user-created MySQL database.

![InnoDB: Shutdown completed...]

---

**Note:**
The Percona XtraBackup version you use is based on your MySQL version:

- MySQL 5.6 and earlier versions require that you install Percona XtraBackup 2.3. For more information, see Percona XtraBackup 2.3.
- MySQL 5.7 requires that you install Percona XtraBackup 2.4. For more information, see Percona XtraBackup 2.4.
14. Edit the parameters in the backup-my.cnf file to ensure version compatibility.

a. Run the following command to modify the backup-my.cnf file in text mode:

```
vi /home/mysql/data/backup-my.cnf
```

b. Comment out the following parameters. The following parameters are not supported by the user-created MySQL database:

```
#innodb_log_checksum_algorithm
#innodb_fast_checksum
#innodb_log_block_size
#innodb_doublewrite_file
#rds_encrypt_data
#innodb_encrypt_algorithm
#redo_log_version
#master_key_id
```

**Note:**

- The MyISAM storage engine is incompatible with the InnoDB storage engine used by ApsaraDB for RDS. If the user-created MySQL database runs MyISAM, you must comment out the following parameters and add the `skip-grant-tables` parameter:

```
#innodb_log_checksum_algorithm=strict_crc32
#redo_log_version=1
skip-grant-tables
```

- If the user-created MySQL database runs MyISAM, and the system reports storage engine-related error messages when you manage a system table, you must run the following command to change the storage engine of the system table:

```
alter engine <The name of the system table> engine=myisam;
```


15. Run the following command to change the owner of the backup-my.cnf file to a user who has permissions to manage the user-created MySQL instance:

```
chown -R mysql:mysql /home/mysql/data
```

16. Run the following command to start the MySQL process:

```
mysqld_safe --defaults-file=/home/mysql/data/backup-my.cnf --user=mysql --datadir=/home/mysql/data &
```

**Note:**

We recommend that you reset the password of the root user. For more information, see How to Reset the Root Password.
17. Run the following command to log on to the user-created MySQL database to verify that the MySQL process has started:

```bash
mysql -uroot -p<The password used to log on to the user-created MySQL database>
```

If the system displays information similar to the following, the MySQL process has started. This means that the required parameters have been commented out and the owner of the backup-my.cnf file has been changed.

```
[root@testcentos ~]# mysql -uroot
Welcome to the MySQL monitor. Commands end with ; or \
Your MySQL connection id is 1
Server version: 5.5.43 MySQL Community Server (GPL) by Remi
Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> 
```

**FAQ**

- In addition to data backup files, which other methods can I use to restore my ApsaraDB RDS for MySQL instance to a user-created MySQL database?

You can use Alibaba Cloud Data Transmission Service (DTS) to migrate your ApsaraDB RDS for MySQL instance to a user-created MySQL database. For more information, see [Migrate data from an ApsaraDB RDS for MySQL database to a user-created MySQL database](#).

- What do I do if the system reports errors when I download a data backup file?

If you use the `wget -c '<The external download URL of the data backup file>' -O <The name to use for the downloaded data backup file>.tar.gz` command to download the data backup file, check that the download URL is enclosed in a pair of single quotation marks ('). This makes it easier for the system to identify the download URL.
• What do I do if the system reports errors when I decompress the data backup file that I downloaded?

1. Check that the data backup file is a physical backup file.
2. Check that the data backup file is saved with a valid extension (.tar.gz, .xb.gz, or _qp.xb).
3. Check that you are running the correct command based on the format of the data backup file. For more information, see Step 11 in the "Procedure" section.

• How do I restore or migrate my ApsaraDB RDS for MySQL instance in the Basic Edition?

Instances in the Basic Edition only support snapshot backups. If your ApsaraDB RDS for MySQL instance is in the Basic Edition, use one of the following two methods to perform a restore or migrate operation:

- Migrate data to an RDS MySQL instance by using mysqldump.
- Use DTS to export data from your ApsaraDB RDS for MySQL instance to your computer.

• Can I restore a data backup file of my ApsaraDB RDS for MySQL instance to another ApsaraDB RDS for MySQL instance?

No, this operation is not supported. We recommend that you use DTS to migrate your ApsaraDB RDS for MySQL instance to another. For more information, see Migrate data between RDS instances.

References

• Restore the data of an RDS MySQL instance
• Restore individual databases and tables of an ApsaraDB RDS for MySQL instance
• Restore an ApsaraDB RDS for MySQL instance across regions

14.6 Restore data from logical backup files of an ApsaraDB RDS MySQL instance to an on-premises database

This topic describes how to restore a database through logical backup files. You can use the mysqldump utility that comes with MySQL to restore a database through logical backup files.

Note:
For more information about how to restore physical backup files to on-premises databases, see Use a physical backup file to restore an ApsaraDB RDS for MySQL instance to a user-created MySQL database.

For more information about how to back up ApsaraDB RDS MySQL data, see #unique_273.

Precautions

The on-premises MySQL database is installed in a 64-bit Linux system. The version of the database is the same as that of the ApsaraDB RDS MySQL instance. This topic describes how to restore data from logical backup files of an ApsaraDB RDS MySQL 5.7 instance running the Linux 7 system.

Prerequisites

The instance is in one of the following editions:

- MySQL 5.7 High-availability Edition (based on local SSDs)
- MySQL 5.6
- MySQL 5.5

Procedure

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target instance is located.
3. Find the instance and click the instance ID.
4. In the left-side navigation pane, click Backup and Restoration.
5. Select a time range and click Search.
6. In the backup list, find the backup file and click Download.

Note:

If Download does not appear, make sure that the version of your instance allows you to download logical backup files.
7. In the **Download Instance Backup Set** dialog box that appears, click **Copy External Download URL**.

![Copy External Download URL](image)

8. Log on to the Linux system where the on-premises database resides and run the following command to download the logical backup files:

```
wget -c '<External download URL>' -O <Custom file name>.tar.gz
```

**Note:**
- `-c`: specifies to resume from the breakpoint.
- `-O`: saves the downloaded results as a specified file.

9. Run the following command to decompress the logical backup files, including the default and user-generated database files:

```
tar xvf <Data backup file name>.tar
```

10. Decompress the compressed files for the database to be restored.

```
gzip-d <Compressed database file name>.gz
```

11. Log on to the on-premises database to create an empty database.

```
mysql -u root -p<Database password>
mysql> create database <Database name>;
Query OK, 1 row affected (0.00 sec)
mysql> exit
```
12. Run the following command to import the .sql file to the empty database:

```
mysql -u root -p <Database password> <Empty database name> <~/<Database name after decompression>.sql
```

13. Log on to the database and check for table data. If table data exists, the migration is successful.
15 Read-only instances

15.1 Set a replication delay for an RDS MySQL read-only instance

This topic describes how to set a replication delay for an RDS MySQL read-only instance. The read-only instance replicates data from the corresponding master instance at the specified delay.

Context

After you set a replication delay for a read-only instance, the read/write splitting function does not take effect on the read-only instance. To make the read/write splitting function take effect on the read-only instance, you must change the replication delay of the read-only instance to 0 seconds.

![Configure Read/Write Splitting](image)

Procedure

1. Log on to the RDS console.
2. Select the target region.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click **Service Availability**.

5. Click **Set Delayed Replication**.

6. In the displayed dialog box, set **Delay** and click **OK**.

**Note:**
The delay is measured in the unit of second. The default delay is 0 seconds, indicating that the master instance sends operation log data to the read-only instance immediately after completing an operation and the read-only instance performs the operation immediately after receiving the operation log data.

### 15.2 Overview of ApsaraDB RDS for MySQL read-only instances

This topic provides an overview of the read-only instances of ApsaraDB RDS for MySQL. If a large number of read requests overwhelm the primary instance, your business may be
interrupted. In this situation, you can add one or more read-only instances to offload read requests from the primary instance and increase the throughput of your application.

For information about read-only instances in other database engines, see the following topics:

- #unique_279
- #unique_280
- #unique_281

**Background information**

ApsaraDB RDS for MySQL replicates data from the secondary instance to every read-only instance you create. After data on the primary instance is updated, the updated data is replicated to each read-only instance. You can set a replication delay for a read-only instance. For more information, see [Set a replication delay for an RDS MySQL read-only instance](#).

**Note:**

Each read-only instance works in a primary/secondary architecture to ensure high availability.

**Prerequisites**

The instance runs one of the following MySQL versions and RDS editions:

- MySQL 8.0 in the High-availability Edition or Enterprise Edition
- MySQL 5.7 in the High-availability Edition or Enterprise Edition
- MySQL 5.6

**Note:**

If your instance running MySQL 5.7 in the Enterprise Edition does not support read-only instances, [submit a ticket](#).

**Billing**

A read-only instance is billed on an hourly basis. The actual fee varies based on the instance type at the time of fee deduction. For more information, visit [ApsaraDB RDS for MySQL pricing](#).

**Features**

- Read-only instances use pay-as-you-go billing to reduce costs.
• Read-only instances reside in the same region as the primary instance, but possibly in different zones.

• The specifications of a read-only instance can differ from the specifications of the primary instance, and can be changed at any time. We recommend that the specifications of a read-only instance be greater than or equal to the specifications of the primary instance. If the specifications of a read-only instance are lower than the specifications of the primary instance, the read-only instance may encounter high latency or be unable to handle heavy loads.

• The network type of a read-only instance can differ from that of the primary instance.

• Read-only instances do not require database or account maintenance, because their database and account information is synchronized with the primary instance.

• ApsaraDB RDS for MySQL automatically replicates the whitelists of the primary instance to every read-only instance you create. After a read-only instance is created, its whitelists are independent of the whitelists configured on the primary instance. For information about how to modify the whitelists of a read-only instance, see Configure a whitelist for an ApsaraDB RDS for MySQL instance.

• Read-only instances support the monitoring and alerting for up to 20 performance metrics such as the disk capacity, IOPS, number of connections, CPU utilization, and network traffic.

**Limits**

• The maximum number of read-only instances allowed varies based on the memory capacity.

<table>
<thead>
<tr>
<th>Database engine</th>
<th>Memory capacity</th>
<th>Maximum number of read-only instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>≥ 64 GB</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>&lt; 64 GB</td>
<td>5</td>
</tr>
</tbody>
</table>

• You cannot configure backup policies or manually create backups for read-only instances, because these are already configured or created on the primary instance.
• Take note of the following limits on instance restoration:
  - You cannot create a temporary read-only instance from a data backup file or a specific point in time, nor can you overwrite a read-only instance by using a data backup file.
  - After a read-only instance is created, you cannot use a data backup file to restore it in overwrite mode.
• You cannot migrate data to read-only instances.
• You cannot create or delete databases on read-only instances.
• You cannot create or delete accounts, authorize accounts, or change the passwords of accounts on read-only instances.

**Instance creation**

For more information, see Create an ApsaraDB RDS for MySQL read-only instance.

**FAQ**

• Can I manage accounts created on a primary instance from its read-only instances?
  
  No, although the accounts created on a primary instance are synchronized to its read-only instances, they cannot be managed on the read-only instances. In addition, the accounts only have read permissions on the read-only instances.

• Can I pause the fee deduction for read-only instances? And can I set the read weights of read-only instances to 0 to stop the billing?

  No, you cannot pause the fee deduction for read-only instances, because they are billed on an hourly basis. However, you can release the read-only instances when you no longer need them. For more information, see Release an RDS MySQL instance.

### 15.3 Create an ApsaraDB RDS for MySQL read-only instance

This topic describes how to create an ApsaraDB RDS for MySQL read-only instance. If the primary instance of your database system is overwhelmed by large amounts of read requests, you can create read-only instances to offload read requests from the primary instance and increase the throughput of your application. Each read-only instance is a replica of the primary instance. The primary instance replicates updates to its data to all read-only instances created in it.

For more information about how to create read-only instances in other database engines, see the following topics:
For more information about read-only instances, see Overview of ApsaraDB RDS for MySQL read-only instances.

**Prerequisites**

The instance runs one of the following MySQL versions and RDS editions:

- MySQL 8.0 in the High-availability Edition or Enterprise Edition
- MySQL 5.7 in the High-availability Edition or Enterprise Edition
- MySQL 5.6

**Note:**
If your instance running MySQL 5.7 in the Enterprise Edition does not support read-only instances, submit a ticket.

**Precautions**

- You can only create read-only instances in the primary instance. You cannot switch existing instances to read-only instances.
- While you create a read-only instance, the system replicates data from a secondary instance. Therefore, the operation of the primary instance is not interrupted.
- A read-only instance does not inherit the parameter settings of the primary instance. The system generates default parameter settings for it. You can only reconfigure the parameters of a read-only instance in the ApsaraDB for RDS console.
- The maximum number of read-only instances you can create in the primary instance varies depending on the memory capacity.

<table>
<thead>
<tr>
<th>Memory capacity</th>
<th>Maximum number of read-only instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>≥ 64 GB</td>
<td>10</td>
</tr>
<tr>
<td>&lt; 64 GB</td>
<td>5</td>
</tr>
</tbody>
</table>

- Each read-only instance is charged at an hourly rate by pay-as-you-go billing. The hourly rate varies depending on the instance type at the time of deduction. For more information, visit ApsaraDB RDS for MySQL.
**Create a read-only RDS instance**

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the Distributed by Instance Role section of the Basic Information page, click **Create Read-Only Instance**.
5. Configure the parameters of the read-only instance.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Zone</strong></td>
<td>The zone where the read-only instance resides. Each zone is an independent physical area located within a region. There are no substantive differences between zones in the same region. If your database system spans multiple zones, it can provide zone-level disaster recovery.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Instance Type | • **Entry-level**: belongs to the general-purpose instance family. A general-purpose instance occupies exclusive memory and I/O resources allocated to it, but shares CPU and storage resources with the other general-purpose instances on the same physical host.  
• **Enterprise-level**: belongs to the dedicated instance family. A dedicated instance occupies exclusive CPU, memory, storage, and I/O resources allocated to it. The top configuration of the dedicated instance family is dedicated host: A dedicated-host instance occupies all CPU, memory, storage, and I/O resources on the physical host where it is housed. |
|              | **Note:** Each instance type supports a specific number of CPU cores, memory capacity, maximum number of connections, and maximum input/output operations per second (IOPS). For more information, see #unique_82. |
| Capacity     | The storage capacity to store the data files, system files, binary log files, and transaction files of the read-only instance. You can adjust the storage capacity at increments of 5 GB.                                                                                       |
|              | **Note:** The dedicated instance family support exclusive allocations of resources. Therefore, the storage capacity of each instance type with local SSDs in this family is fixed and cannot be adjusted. For more information, see #unique_82. |
6. Click **Next: Instance Configuration** and configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Type</td>
<td>• <strong>Classic Network</strong>: a traditional type of network.</td>
</tr>
<tr>
<td></td>
<td>• <strong>VPC</strong>: A VPC is an isolated virtual network with higher security and</td>
</tr>
<tr>
<td></td>
<td>better performance than a classic network. If you select the VPC network</td>
</tr>
<tr>
<td></td>
<td>type, you must also specify <strong>VPC</strong> and <strong>VSwitch of Primary Node</strong>.</td>
</tr>
</tbody>
</table>

**Note:**
Make sure that the primary RDS instance and the ECS instance you want to connect have the same network type. If both their network types are VPC, you must also make sure that they reside in the same VPC. Otherwise, they cannot communicate over an internal network.

7. Click **Next: Confirm Order**.

8. Read and select Terms of Service and click **Pay Now**.

The read-only instances are created after a few minutes.

**View a read-only instance**

To view a read-only instance on the Instances page, follow these steps:

1. Log on to the ApsaraDB for RDS console.
2. Select the region where the read-only instance resides.
3. Find the read-only instance and click its ID.

To view a read-only instance on the Basic Information page of its primary RDS instance, follow these steps:

1. Log on to the ApsaraDB for RDS console.
2. Select the region where the primary RDS instance resides.
3. Find the primary RDS instance and click its ID.
4. On the **Basic information** page, move the pointer over the number of read-only instances and click the ID of the read-only instance you want to view.

![Basic Information](image)

**View the delay of data replication to a read-only instance**

A primary RDS instance replicates data to its read-only instances at a certain delay. You can navigate to the Basic Information page of a read-only instance to view the delay of data replication to it.

![Delay](image)

**Related operations**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_285</td>
<td>Creates a read-only instance for a primary ApsaraDB for RDS instance.</td>
</tr>
</tbody>
</table>

**FAQ**

- Will the system automatically upgrade the specifications of read-only instances after I upgrade the specifications of their primary RDS instance?
  
  No, you must manually upgrade the specifications of read-only instances.

- Can I access a read-only instance by using a read/write splitting address after I set its read weight to 0?
  
  No, you can only access a read-only instance by using its own **internal or public endpoint** after you set its read weight to 0. We recommend that you only set the read weight of a read-only instance to 0 if you want this instance to serve only a specific service.
16 Database Autonomy Service

16.1 DAS overview

DAS (Database Autonomy Service) is a cloud service that uses machine learning and expert experience to implement database self-detection, self-repair, self-optimization, self-maintenance, and self-security. It helps you eliminate the complexity of database management and service failures caused by manual operations, thus effectively ensuring the stability, security, and efficiency of database services.

Overview

The MySQL autonomy service has the following features:

- **Sessions Management**
  You can view sessions, collect session statistics, analyze SQL statements, and optimize the execution of SQL statements.

- **Real-time Monitoring**
  You can view real-time information such as the queries per second (QPS), transactions per second (TPS), number of connections, and network traffic.

- **Storage Analysis**
  You can view the space utilization, trend, exceptions, tablespaces, and data spaces.

- **Performance insight**
  This function helps you evaluate database loads and find the root causes of performance problems to make your instance more stable.

- **Dashboard**
  You can view and compare performance trends, customize monitoring dashboards, detect exceptions, and view the instance topology.

- **Slow Query Logs**
  You can view the trends and statistics of slow SQL statements.

- **Report**
  This function provides both manual and automatic diagnosis for you to view the instance health, alerts, and slow SQL statements.
16.2 Diagnostics

In ApsaraDB RDS for MySQL, DAS provides the diagnostics feature. This feature diagnoses your ApsaraDB RDS for MySQL instance and visualizes the results.

Prerequisites

Your RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 8.0 in the High-availability or Enterprise Edition
- MySQL 5.7 in the High-availability or Enterprise Edition
- MySQL 5.6
- MySQL 5.5

Open the Diagnostics page

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where your RDS instance resides.
3. Click the ID of your RDS instance.
4. In the left-side navigation pane, choose Database Autonomy Service (CloudDBA) > Diagnostics.
5. Click the Diagnostics tab.

Note:
For more information, see Diagnostics.
16.3 Autonomy center

In ApsaraDB RDS for MySQL, DAS provides the autonomy center feature. If DAS detects an exception on core metrics, it automatically diagnoses sessions, SQL statements, and database capacity to identify possible causes. DAS also provides optimization and mitigation suggestions. However, DAS does not implement the suggestions until you grant it permissions.

Prerequisites

Your RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 8.0 in the High-availability or Enterprise Edition
- MySQL 5.7 in the High-availability or Enterprise Edition
- MySQL 5.6
- MySQL 5.5

Open the Autonomy Center page

1. Log on to the ApsaraDB for RDS console.

2. In the top navigation bar, select the region where your RDS instance resides.

3. Click the ID of your RDS instance.

4. In the left-side navigation pane, choose Database Autonomy Service (CloudDBA) > Diagnostics.
5. Click the **Autonomy Center** tab.

![Autonomy Center tab](image)

**Note:**
For more information, see **Autonomy center**.

### 16.4 Sessions Management

DAS provides the session management function for RDS MySQL, allowing you to view and manage sessions for the target instance.

**Prerequisites**

The MySQL version and RDS edition are as follows:

- MySQL 8.0 High-availability Edition or Enterprise Edition
- MySQL 5.7 High-availability Edition or Enterprise Edition
- MySQL 5.6
- MySQL 5.5

**Go to the session management page**

1. Log on to the **RDS console**.
2. In the upper-left corner of the page, select the region where the RDS instance resides.
3. Find the target RDS instance and click the instance ID.
4. In the left-side navigation pane, choose **autonomous Service (formerly CloudDBA)** > **one-click diagnosis**.
5. Click **session Management** tab.

<table>
<thead>
<tr>
<th>Autonomy Center</th>
<th>Session Management</th>
<th>Real-time Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DAS</strong> (Data Update Time: Jun 15, 2020, 15:21:30)</td>
<td>Select a session before you click Optimize.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
For more information, see View instance sessions.

### 16.5 Real-time Monitoring
DAS provides the real-time monitoring function for RDS MySQL to view the real-time performance of MySQL.

**Prerequisites**
The MySQL version and RDS edition are as follows:

- MySQL 8.0 High-availability Edition or Enterprise Edition
- MySQL 5.7 High-availability Edition or Enterprise Edition
- MySQL 5.6
- MySQL 5.5

**Go to the real-time monitoring page**

1. Log on to the RDS console.
2. In the upper-left corner of the page, select the region where the RDS instance resides.
3. Find the target RDS instance and click the instance ID.
4. In the left-side navigation pane, choose **autonomous Service (formerly CloudDBA)** > one-click diagnosis.
5. Click the **Real-time Monitoring** tab.

<table>
<thead>
<tr>
<th>Session Management</th>
<th>Real-time Monitoring</th>
<th>Storage Analysis</th>
</tr>
</thead>
</table>

**Note:**
For more information, see real-time performance.

### 16.6 Storage Analysis

DAS provides the storage analysis function for RDS MySQL. This function can help you detect database space exceptions in a timely manner and ensure database stability.

**Prerequisites**

The MySQL version and RDS edition are as follows:

- MySQL 8.0 High-availability Edition or Enterprise Edition
- MySQL 5.7 High-availability Edition or Enterprise Edition
- MySQL 5.6
- MySQL 5.5

**Context**

The storage analysis function allows you to view the space usage of a database instance, the number of days the space is available, and the space usage, space fragmentation, and space exception diagnosis of a table in the database.

**Go to the storage analysis page**

1. Log on to the RDS console.
2. In the upper-left corner of the page, select the region where the RDS instance resides.
3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, choose autonomous Service (formerly CloudDBA) > one-click diagnosis.

5. Click spatial analysis tab.

![Real-time Monitoring | Storage Analysis | Capacity Assessment](image)

<table>
<thead>
<tr>
<th>Note:</th>
</tr>
</thead>
<tbody>
<tr>
<td>For more information, see spatial analysis.</td>
</tr>
</tbody>
</table>

### 16.7 Capacity assessment

In ApsaraDB RDS for MySQL, DAS provides the capacity assessment feature. This feature allows you to view capacity suggestions, performance capacity, storage usage, and the service time of available storage. It also uses machine learning and capacity algorithms to forecast storage usage.

**Prerequisites**

Your RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 8.0 in the High-availability or Enterprise Edition
- MySQL 5.7 in the High-availability or Enterprise Edition
- MySQL 5.6
- MySQL 5.5

**Open the Capacity Assessment page**

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where your RDS instance resides.

3. Click the ID of your RDS instance.

4. In the left-side navigation pane, choose **Database Autonomy Service (CloudDBA)** > **Diagnostics**.

5. Click the **Capacity Assessment** tab.

<table>
<thead>
<tr>
<th>Storage Analysis</th>
<th><strong>Capacity Assessment</strong></th>
<th>Deadlock Analysis</th>
</tr>
</thead>
</table>

**Note:**
For more information, see [Capacity assessment](#).

### 16.8 Deadlock analysis

In ApsaraDB RDS for MySQL, DAS provides the deadlock analysis feature. This feature allows you to view and analyze the last deadlock in a database.

**Prerequisites**

Your RDS instance runs one of the following MySQL versions and RDS editions:

- MySQL 8.0 in the High-availability or Enterprise Edition
- MySQL 5.7 in the High-availability or Enterprise Edition
- MySQL 5.6
- MySQL 5.5

**Open the Deadlock Analysis page**

1. Log on to the [ApsaraDB for RDS console](#).
2. In the top navigation bar, select the region where your RDS instance resides.

3. Click the ID of your RDS instance.

4. In the left-side navigation pane, choose **Database Autonomy Service (CloudDBA)** > **Diagnostics**.

5. Click the **Deadlock Analysis** tab.

---

**Note:**
For more information, see [Deadlock analysis](#).

16.9 **Performance insight**

Performance Insight, which is used in DAS to monitor the load, perform association analysis, and optimize the performance of RDS instances. It helps you evaluate the database load in a simple and intuitive way. Find the source of performance problems to improve database stability.

**Prerequisites**

The MySQL version and RDS edition are as follows:

- MySQL 8.0 High-availability Edition or Enterprise Edition
- MySQL 5.7 High-availability Edition or Enterprise Edition
- MySQL 5.6
- MySQL 5.5
Go to the performance insight page

1. Log on to the RDS console.

2. In the upper-left corner of the page, select the region where the RDS instance resides.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, choose Autonomy Service (CloudDBA) > Performance insight.

Note:
For more information, see Performance Insight.

16.10 Dashboard

DAS provides the dashboard function. This function allows you to view performance trends in multiple ways. In addition to the performance trend interval, DAS also supports performance trend comparison and custom performance trend viewing.

Prerequisites

The MySQL version and RDS edition are as follows:

- MySQL 8.0 High-availability Edition or Enterprise Edition
- MySQL 5.7 High-availability Edition or Enterprise Edition
- MySQL 5.6
- MySQL 5.5

Go to the dashboard page

1. Log on to the RDS console.
2. In the upper-left corner of the page, select the region where the RDS instance resides.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, choose Autonomy Service (CloudDBA) > Dashboard.

**Note:**
For more information, see Performance trends.

**16.11 Slow Query Logs**

The slow query logs feature of ApsaraDB RDS for MySQL allows you to view the trends, execution status, and optimization suggestions of slow query logs.

**Prerequisites**

The MySQL version and RDS edition are as follows:

- MySQL 8.0 High-availability Edition or Enterprise Edition
- MySQL 5.7 High-availability Edition or Enterprise Edition
- MySQL 5.6
- MySQL 5.5

**Go to the slow query logs page**

1. Log on to the RDS console.
2. In the upper-left corner of the page, select the region where the RDS instance resides.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, choose **Autonomy Service (CloudDBA) > Slow Query Logs**.

   **Note:**
   For more information, see **Diagnostics**.

### 16.12 Full SQL statistics

In ApsaraDB RDS for MySQL, DAS provides the full SQL statistics feature. This feature allows you to view and compare the number of executions, execution duration, and execution duration distribution of each SQL statement within a specific time range. This allows you to identify problem SQL statements.

**Prerequisites**

- Your RDS instance runs one of the following MySQL versions and RDS editions:
  - MySQL 8.0 in the High-availability or Enterprise Edition
  - MySQL 5.7 in the High-availability or Enterprise Edition
  - MySQL 5.6
  - MySQL 5.5
- SQL Explorer is enabled for your RDS instance. For more information, see **SQL Explorer**.

**Note:**
You can also follow the instructions in the ApsaraDB for RDS console to enable SQL Explorer.

**Context**
ApsaraDB for RDS

The full SQL statistics feature is based on SQL Explorer. This feature has the Basic Edition and Professional Edition.

- If your SQL Explorer is of the Trial Edition, the full SQL statistics feature must be of the Basic Edition. You can only view the number of executions and the execution duration of each SQL statement.
- If your SQL Explorer is of the Paid Edition, the full SQL statistics feature must be of the Professional Edition. The Professional Edition provides the following functions:
  - Optimized data analysis and fast responses
  - Comparative data display and source statistics
  - Detection of dangerous SQL statements, SQL injection, and new access sources

Billing

Currently, you can use the full SQL statistics feature for free, but you must pay for the storage of audit logs generated by SQL Explorer. If this feature is billed after its public preview ends, you will receive notifications.

Precautions

- In the Professional Edition, the intervals for Display by Time Range and Source Statistics cannot exceed three days. In the Basic Edition, the interval for Full Request Analysis cannot exceed one day.
- Data is updated at 10-minute intervals.

Open the Full SQL Statistics page

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where your RDS instance resides.
3. Click the ID of your RDS instance.
4. In the left-side navigation pane, choose **Database Autonomy Service (CloudDBA) > Full SQL Statistics**.

**Note:**
For more information, see **Full SQL statistics**.

### 16.13 Report

DAS provides the RDS for MySQL report function, allowing you to create and view diagnostic reports.

**Prerequisites**

The MySQL version and RDS edition are as follows:

- MySQL 8.0 High-availability Edition or Enterprise Edition
- MySQL 5.7 High-availability Edition or Enterprise Edition
- MySQL 5.6
- MySQL 5.5

**Go to the reports page**

1. Log on to the **RDS console**.

2. In the upper-left corner of the page, select the region where the RDS instance resides.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, choose **Autonomy Service (CloudDBA) > Report**.

**Note:**
For more information, see **Diagnostic Report**.
17 Monitoring and alerts

17.1 View resource and engine monitoring data

This topic describes how to view the resource and engine monitoring data of an RDS for MySQL instance. ApsaraDB for RDS provides a wide range of performance metrics for you to view in the RDS console.

Procedure

1. Log on to the RDS console.
2. In the upper-left corner, select the region where the target RDS instance is located.
3. Find the target RDS instance and click the instance ID.
4. In the left-side navigation pane, click Monitoring and Alerts.
5. On the Monitoring tab, select the Resource Monitoring or Engine Monitoring monitoring type and specify the time range. The following table describes the monitoring metrics.

<table>
<thead>
<tr>
<th>Monitoring type</th>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource</td>
<td>Disk Space (MB)</td>
<td>The disk space usage of the RDS instance, including:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Instance Size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Data Usage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Log Size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Temporary File Size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Other System File Size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unit: MByte.</td>
</tr>
<tr>
<td>Engine</td>
<td>IOPS (Input/Output Operations</td>
<td>The number of I/O requests per second for the RDS instance. Unit: Number/second.</td>
</tr>
<tr>
<td></td>
<td>per Second)</td>
<td></td>
</tr>
</tbody>
</table>

Issue: 20200702
<table>
<thead>
<tr>
<th>Monitoring type</th>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitoring</td>
<td>Total Connections</td>
<td>The total number of connections to the RDS instance, including the number of active connections and the total number of connections.</td>
</tr>
<tr>
<td></td>
<td>CPU and Memory Usage (%)</td>
<td>The CPU and memory usage of the RDS instance (excluding the CPU and memory usage for the operating system).</td>
</tr>
<tr>
<td></td>
<td>Network Traffic (KB)</td>
<td>The input and output traffic of the RDS instance per second. Unit: KB.</td>
</tr>
<tr>
<td>Engine Monitoring</td>
<td>TPS (Transactions per Second)/QPS (Queries per Second)</td>
<td>The average number of transactions per second and the average number of SQL statements executed per second.</td>
</tr>
<tr>
<td></td>
<td>InnoDB Buffer Pool Read Hit Ratio, Usage Ratio, and Dirty Block Ratio (%)</td>
<td>The read hit ratio, usage, and proportion of dirty blocks for the InnoDB buffer pool.</td>
</tr>
<tr>
<td></td>
<td>InnoDB Read/Write Volume (KB)</td>
<td>The amount of data that is read and written by InnoDB per second. Unit: KB.</td>
</tr>
<tr>
<td></td>
<td>InnoDB Buffer Pool Read/Write Frequency</td>
<td>The number of read and write operations that are performed by InnoDB per second.</td>
</tr>
<tr>
<td></td>
<td>InnoDB Log Read/Write/fsync</td>
<td>The average frequency of physical writes to log files per second by InnoDB, the log write request frequency, and the average frequency of fsync() writes to log files.</td>
</tr>
<tr>
<td></td>
<td>Number of Temporary Tables Created Automatically on the Hard Disk when MySQL Statements Are Being Executed</td>
<td>The number of temporary tables that are automatically created on the hard disk when the instance runs SQL statements.</td>
</tr>
<tr>
<td>Monitoring type</td>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>MySQL_COMDML</td>
<td>The number of SQL statements that are executed by the RDS instance per second, including:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Insert</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Delete</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Insert_Select</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Replace</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Replace_Select</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Select</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Update</td>
</tr>
<tr>
<td></td>
<td>MySQL_RowDML</td>
<td>The number of operations that are performed by InnoDB per second, including:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The average number of physical writes to log files per second</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The number of rows that are read/updated/deleted/inserted from InnoDB tables per second.</td>
</tr>
<tr>
<td></td>
<td>MyISAM Read/Write Frequency</td>
<td>The number of read/write operations that are performed by MyISAM on the buffer pool per second and the number of read/write operations that are performed by MyISAM on the hard disk per second.</td>
</tr>
<tr>
<td></td>
<td>MyISAM Key Buffer Read/Write/Usage Ratio (%)</td>
<td>The read hit ratio, write hit ratio, and usage of the MyISAM key buffer per second.</td>
</tr>
</tbody>
</table>

### 17.2 Set the monitoring frequency of an ApsaraDB RDS for MySQL instance

This topic describes how to set the monitoring frequency of an ApsaraDB RDS for MySQL instance.

**Background information**

Before May 15, 2018, ApsaraDB for RDS provided the following two monitoring frequencies:

- Every 60 Seconds: The monitoring period is 30 days.
- Every 300 Seconds: The monitoring period is 30 days.

As your database system expands, these two monitoring frequencies may not meet your business needs.
In response to customer needs, ApsaraDB for RDS now provides second-level monitoring to help O&M engineers obtain finer-grained monitoring data and locate complex database O&M problems.

Every 5 Seconds: The monitoring period is seven days. After the seventh day, 5-second monitoring is downgraded to 60-second monitoring.

The following table describes monitoring policies.

<table>
<thead>
<tr>
<th>Instance specifications</th>
<th>Every 5 Seconds</th>
<th>Every 60 Seconds</th>
<th>Every 300 Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-availability Edition with local SSDs or Enterprise Edition, with a memory capacity less than 8 GB</td>
<td>Not supported</td>
<td>Supported for free</td>
<td>Enabled by default</td>
</tr>
<tr>
<td>High-availability Edition with local SSDs or Enterprise Edition, with a memory capacity greater than or equal to 8 GB</td>
<td>Supported with charges</td>
<td>Enabled by default</td>
<td>Supported for free</td>
</tr>
<tr>
<td>High-availability Edition with standard SSDs or enhanced SSDs (ESSDs)</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Enabled by default</td>
</tr>
<tr>
<td>Basic Edition with MySQL 8.0</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Enabled by default</td>
</tr>
<tr>
<td>Basic Edition with MySQL 5.7</td>
<td>Not supported</td>
<td>Supported for free</td>
<td>Enabled by default</td>
</tr>
</tbody>
</table>

Procedure

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID.

4. In the left-side navigation pane, click **Monitoring and Alerts**.

   ![Monitoring and Alerts](image)

   **Note:**
   For more information about the metrics supported by ApsaraDB RDS for MySQL, see [View resource and engine monitoring data](#).

5. Click the **Monitoring** tab.

6. Click **Set Monitoring Frequency**.

7. In the **Set Monitoring Frequency** dialog box, select a monitoring frequency and click **OK**.

   For this example, select **Every 5 Seconds**.

   ![Set Monitoring Frequency](image)

8. In the **Confirm Changes** message, click **Confirm**.

   If the RDS instance does not support the Every 5 Seconds monitoring frequency, a message appears in the **Set Monitoring Frequency** dialog box.

**FAQ**

**Related operations**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_301</td>
<td>Queries the monitoring frequency of an ApsaraDB RDS for MySQL instance.</td>
</tr>
</tbody>
</table>
17.3 Set Alarm Rules

You can set alert rules for monitoring your system. When exceptions are detected and alert rules triggered, a notification will be sent to the alert contacts.

For more information about how to set alarm rules for other engines, see the following topics:

- Configure alert rules for an ApsaraDB RDS for SQL Server instance
- Configure alert rules for an ApsaraDB RDS for PostgreSQL instance
- Configure alert rules for an ApsaraDB RDS for PPAS instance
- Configure alert rules for an ApsaraDB RDS for MariaDB TX

Prerequisites

The instance must be in the mainland China region.

Background

Monitoring and alerting are implemented through CloudMonitor. With CloudMonitor, you can configure monitored metrics and alert rules, and can associate contact groups with monitored metrics. When a monitored metric triggers an alert based on a specified alert rule, the system sends notification emails to all contacts in the contact groups associated with the monitored metric.

Create alert rules

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click Monitoring and Alerts.
5. On the Monitoring and Alerts page, click the Alerts tab.
6. Click **Set Alert Rule** to go to the CloudMonitor console.

7. Create an alert contact group. For more information, see *Create an alert contact and an alert Contact Group*.

8. Create an alert rule. For more information, see *Apsaradb for RDS*.

**Note:**
You can also use tags to automatically monitor resources. For more information, see *#unique_306*.

**Manage alert rules**

1. Log on to the CloudMonitor console for monitoring ApsaraDB for RDS.
2. Select the region where your instance is located.
3. Find the target RDS instance and click its ID.
4. On the **Alarm Rules** tab, find the target alert rule and select one of the following operations:
   - View: view the details of an alert rule.
   - Alarm Logs: view the alert history for a certain period of time.
   - Modify: modify alert rules. For more information about the parameters, see *Alarm rule parameters*.
   - Disable: disable the selected alert rules. If an alert rule is disabled, no alert is triggered even though the monitored metric meets the conditions.
   - Delete: delete the selected alert rules. An alert rule cannot be restored after you delete it. You can only add it again.
18 Database connection

18.1 Connect to an RDS MySQL instance

This topic describes how to connect to an RDS MySQL instance. After completing the initial configuration, you can connect to your RDS instance from an ECS instance or your computer.

For information about how to connect to your RDS instance in other DB engines, see the following resources:

• #unique_100
• #unique_101
• #unique_102
• #unique_103

Background information

After you create an instance, configure a whitelist, and create an account, you can use DMS, a database client, or CLI to connect to your RDS instance. You can also set the IP address, port, and account information in applications to connect.

Use DMS to connect to an RDS instance

DMS is a graphical data management service provided by Alibaba Cloud. It can be used to manage non-relational databases and relational databases, and supports data and schema management, user authorization, security audit, data trends, data tracking, BI charts, and performance and optimization.

For more information, see Use DMS to log on to an ApsaraDB for RDS instance.

Use a database client to connect to an RDS instance

ApsaraDB RDS MySQL is fully compatible with MySQL. You can connect to an RDS instance from any general-purpose database client in the similar way you connect to a MySQL database. This section describes how to use HeidiSQL to connect to an RDS instance.

1. Start HeidiSQL.
2. In the lower-left area of the Session manager dialog box, click New.
3. Enter the information of the RDS instance to be connected. The following table describes the parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Network type</strong></td>
<td>The method of connecting to the RDS instance. Select <strong>MariaDB</strong> or <strong>MySQL (TCP/IP)</strong>.</td>
</tr>
<tr>
<td><strong>Hostname/IP</strong></td>
<td>Enter the private or public IP address of the RDS instance.</td>
</tr>
<tr>
<td></td>
<td>• If your database client is deployed in an ECS instance that is in the same region and has the same network type as the RDS instance, you can use the private IP address of the RDS instance. For example, if the ECS and RDS instances are both in a VPC located in the China (Hangzhou) region, then you can use the private IP address of the RDS instance to create a secure, efficient connection.</td>
</tr>
<tr>
<td></td>
<td>• In the other situations, use the public IP address of the the RDS instance.</td>
</tr>
<tr>
<td><strong>User</strong></td>
<td>The username of the account that you use to access the RDS instance.</td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>The password of the account that you use to access the RDS instance.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Port</td>
<td>The port for the RDS instance to establish a connection. If you use the private IP address of the RDS instance to establish a connection, enter the private port number. If you use the public IP address of the RDS instance to establish a connection, enter the public port number.</td>
</tr>
</tbody>
</table>

4. Click **Open**.

If the entered information is correct, the RDS instance can be connected.
### Use the CLI to connect to an RDS instance

If MySQL is installed on your server, you can use the CLI to connect to an RDS instance as follows:

```bash
mysql -h<Host name> -P<Port number> -u<Username> -p<Password> -D<RDS instance name>
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>The private or public IP address of the RDS instance. For more information, see #unique_105.</td>
<td>rm-bpxxxxxxxxxxxxx.mysql.rds.aliyuncs.com</td>
</tr>
</tbody>
</table>
| -P    | The port for the RDS instance to establish a connection.  
  - If you use the private IP address of the RDS instance to establish a connection, enter the private port number.  
  - If you use the public IP address of the RDS instance to establish a connection, enter the public port number. | 3306 |
| -u    | The username of the account that you use to access the RDS instance. | root |

**Note:**
- The default port number is 3306.  
- If the port used by the RDS instance to establish a connection is Port 3306, you can retain the default value.
### Field | Description | Example
--- | --- | ---
-p | The password of the account that you use to access the RDS instance. | password233

**Note:**
This field is optional.
- If you do not enter the password in this field, the system prompts you to enter the password during subsequent operations.
- If you enter the password in this field, note that no spaces are allowed between -p and the entered password.

-D | The name of the RDS instance you want to access. | mysql

**Note:**
- This field is optional.
- You can enter only the RDS instance name with -D removed.

#### Figure 18-1: Example of connecting to an RDS instance through CLI
```
[root@]# mysql -p -h myrds -P 3306
Welcome to the MariaDB monitor. Commands end with ; or "."
Your MariaDB host is "myrds" Version: 10.7.25-log Source distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help' or '\h' for help. Type '\c' to clear the current input statement.
MySQL [myrds]> 
```

## 18.2 Apply for a public endpoint for an RDS MySQL instance

This topic describes how to apply for a public endpoint for an RDS MySQL instance. Apsara for RDS supports two types of endpoints: internal endpoints and public endpoints. By default, the system provides you with an internal endpoint for connecting to your RDS instance. If you want to connect to your RDS instance through the Internet, you must apply for a public endpoint.
### Internal and public endpoints

<table>
<thead>
<tr>
<th>Endpoint type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal endpoint</td>
<td>The internal endpoint is generated by default. Use the internal endpoint if all of the following conditions are met:</td>
</tr>
<tr>
<td></td>
<td>• Your application is deployed on an ECS instance.</td>
</tr>
<tr>
<td></td>
<td>• The ECS instance is located in the same region as your RDS instance.</td>
</tr>
<tr>
<td></td>
<td>• The ECS instance has the same network type as your RDS instance.</td>
</tr>
<tr>
<td></td>
<td>The internal endpoint is recommended because accessing RDS through the intranet is most secure and delivers optimal performance.</td>
</tr>
<tr>
<td>Public endpoint</td>
<td>You must manually apply for a public endpoint. You can also release it at anytime.</td>
</tr>
<tr>
<td></td>
<td>Use the public endpoint if you cannot access RDS through the intranet.</td>
</tr>
<tr>
<td></td>
<td>Specific scenarios are as follows:</td>
</tr>
<tr>
<td></td>
<td>• An ECS instance accesses your RDS instance but the ECS instance is located in a different region or has a network type different from your RDS instance.</td>
</tr>
<tr>
<td></td>
<td>• A server or computer outside Alibaba Cloud accesses your RDS instance.</td>
</tr>
</tbody>
</table>

**Note:**
- The public endpoint and traffic are currently free of charge.
- Using the public endpoint reduces security. Please exercise caution.
- To guarantee high security and performance, we recommend that you migrate your application to an ECS instance that is in the same region and has the same network type as your RDS instance and then use the public endpoint.

### Apply for a public endpoint

1. Log on to the [RDS console](https://console.aliyun.com/).

---

ApsaraDB for RDS

RDS MySQL Database / 18 Database connection

438

Issue: 20200702
2. Select the target region.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click **Database Connection**.

5. Click **Apply for Public Endpoint**.

6. In the displayed dialog box, click **OK**. The public endpoint is generated.

7. Optional. To modify the public endpoint or port number, click **Change Endpoint**. In the displayed dialog box, select a connection type, set the public endpoint and port number, and click **OK**.

   - **Connection Type**: Select **Public Endpoint**.

---

**Note:**
The **Public Endpoint** option is available only after you have applied for a public endpoint.

- **Endpoint**: The endpoint must be 8 to 64 characters in length and can contain letters, numbers, and hyphens (-). It must start with a lowercase letter.

- **Port**: You can change the port number only when the network type of the RDS instance is classic network.

### Change Endpoint

<table>
<thead>
<tr>
<th>Connection Type</th>
<th>Endpoint</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Endpoint</td>
<td>rm-xxxxxxxxx.sqlserver.rds.aliyun.com</td>
<td>3433</td>
</tr>
</tbody>
</table>

Starts with a lower-case letter, consists of 8 to 64 characters, including letters, digits, or hyphen (-).

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_309</td>
<td>Used to apply for a public endpoint for an RDS instance.</td>
</tr>
</tbody>
</table>

#### 18.3 Use DMS to log on to an ApsaraDB for RDS instance

This topic describes how to log on to an ApsaraDB for RDS instance by using Alibaba Cloud Data Management (DMS).

**Context**

DMS is an all-in-one data management service that supports data management, structure management, account authorization, security audit, data trends, data tracking, BI charts, performance trends and optimization, and server management.
To improve data management experience, Alibaba Cloud provides a new version of DMS that includes more functions. In addition, the price of the DMS Enterprise edition is lowered. For more information, see DMS pricing model changes.

**Use the new version of DMS to log on to an RDS instance**

**Prerequisites**

- You have an Alibaba Cloud account or have applied for the credentials of a RAM user who has permissions on the target database on your RDS instance. For more information about how to apply for permissions, see Manage permissions.
- Your RDS instance is registered with DMS by the administrator. For more information, see Register an ApsaraDB instance.

1. Log on to the DMS console.
2. In the left-side navigation pane, choose your RDS instance and click Please login first.

**Note:**

If your RDS instance uses the Security Collaboration control mode, you do not need to enter the username and password of the account used for the logon. You can choose Logon-free instance and double-click the target database to log on to that database.
3. In the dialog box that appears, enter the username and password of the account used for the logon and click **OK**.

**Note:**
The account used for the logon must have permissions on the target database. Otherwise, the target database is not displayed in the left-side navigation pane. For more information about how to modify the permissions of an account, see [Modify the permissions of an account for an RDS instance](#).

4. In the left-side navigation pane, choose **Logged in instance** and double-click the target database to switch to that database.

**Use the original version of DMS to log on to an RDS instance**

**Note:**
If your DMS has been upgraded to the new version, we recommend that you use the new version of DMS to log on to your RDS instance.

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where your RDS instance resides.

3. Find your RDS instance and click its ID to open the **Basic Information** page.
4. In the upper-right corner of the page, click **Log On to DB** to open the RDS Database Logon page.

**Note:**
Alibaba Cloud directs you to the original or new version of DMS based on your previous logon behavior.
5. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Endpoint:Port number</strong></td>
<td>The endpoint and port number that are used to connect to your RDS instance. The endpoint and port number are in the &lt;Endpoint&gt;:&lt;Port number&gt; format. Example: bpxxxxxxx.rds.aliyuncs.com:3433. For more information about how to view the endpoint and port number of an RDS instance, see <a href="#">View and modify the internal and public endpoints and ports of an ApsaraDB for RDS instance</a>.</td>
</tr>
<tr>
<td><strong>Database Username</strong></td>
<td>The username of the account that is used to connect to your RDS instance.</td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>The password of the account that is used to connect to your RDS instance.</td>
</tr>
</tbody>
</table>

6. Click **Log On**.

**Note:**
If you want the browser to remember the password, select **Remember Password** before you click **Log On**.
7. If the system prompts you to add the Classless Inter-Domain Routing (CIDR) block that contains the IP address of the DMS server to a whitelist of your RDS instance, click **Specify for All Instances** or **Specify for Current Instance**.

8. Click **Log On**.

### 18.4 View and modify the internal and public endpoints and ports of an ApsaraDB for RDS instance

When you connect to an RDS instance, you must enter the internal and public endpoints and their corresponding port numbers. This topic describes how to view and modify the internal and public endpoints and ports of an ApsaraDB for RDS instance in the ApsaraDB for RDS console.

**View the internal and public endpoints and ports**

1. Log on to the ApsaraDB for RDS console.

2. In the upper-left corner of the page, select the region where the target instance is located.

3. Find the target RDS instance and click its ID. The **Basic Information** page appears.

4. On the **Basic Information** page that appears, view the internal and public endpoints and their corresponding port numbers, as shown in the following figure.

**Note:**

- You must configure the whitelist of the instance to display the endpoint information.
Only after you have applied for a public endpoint, you can view this information on the Basic Information page.

Modify the internal and public endpoints and ports

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the instance is located.
3. Find the target RDS instance and click the instance ID.
4. In the left-side navigation pane, click Database Connection.
5. In the upper-right corner of the Database Connection section, click Change Endpoint.
6. In the dialog box that appears, specify Connection Type, modify the endpoint and port, and click OK.

Note:

- The prefix of the endpoint must be 8 to 64 characters in length and can contain letters, digits, and hyphens (-). It must start with a lowercase letter.
The port number must be within the range of 1000 to 5999.

FAQ

- Do I need to modify the endpoint or port in my application after I modify the endpoint or port of an instance?

  You must modify the endpoint or port in the application after you have modified them. Otherwise, the application cannot connect to databases of the instance.

- Does the modification of the endpoint or port take effect immediately? Do I need to restart the instance?

  You do not need to restart the instance. The modification takes effect immediately.

- Can I use the original endpoint of an instance in another instance after the original endpoint is modified or released?

  Yes.

References

For more information about how to enable, modify, and disable a proxy endpoint, see Dedicated proxy.
18.5 Configure a hybrid access solution to smoothly migrate an RDS instance from the classic network to a VPC

This topic describes how to configure a hybrid access solution to smoothly migrate an RDS instance from the classic network to a VPC. To meet the increasing needs of migration between different network types, ApsaraDB for RDS introduces the hybrid access solution. This solution enables a smooth migration from the classic network to a VPC without any transient disconnections or service interruptions. The solution also offers the option to migrate a primary instance and its read-only instances separately without any interference with each other.

Background information

In the past, when migrating an RDS instance from the classic network to a VPC, the internal endpoint of the RDS instance changes. The connection string of the RDS instance remains the same but the IP address bound to the connection string is changed to the corresponding IP address in the VPC. This change will cause a 30-second transient disconnection, and the ECS in the classic network cannot access the RDS instance through the internal endpoint within this period. To migrate the RDS instance across different networks in a smooth manner, ApsaraDB for RDS introduces the hybrid access solution.

Hybrid access refers to the ability of an RDS instance to be accessed by ECS on both the classic network and VPC. During the hybrid access period, the RDS instance reserves the original internal endpoint of the classic network and adds an internal endpoint of VPC. This prevents transient disconnections during the RDS database migration.

For better security and performance, we recommend that you use the internal endpoint of VPC only. Therefore, hybrid access is available for a limited period of time. The internal endpoint of the classic network is released when the hybrid access period expires. In that case, your applications cannot access the RDS database by using the internal endpoint of the classic network. You must configure the internal endpoint of VPC in all your applications during the hybrid access period. This can guarantee smooth network migration and minimize the impact on your services.

For example, your company wants to use the hybrid access solution to migrate RDS instances from the classic network to a VPC. During the hybrid access period, some applications can access the database through the internal endpoint of the VPC, and the other applications can access the database through the original internal endpoint of the classic network. When all the applications access the database through the internal
endpoint of the VPC, the internal endpoint of the classic network can be released. The following figure illustrates the scenario.

**Limits**

During the hybrid access period, the instance has the following limits:

- Switching to the classic network is not supported.
- Migrating the RDS instance to another zone is not supported.

**Prerequisites**

- The network type of the instance is the classic network.
- Available VPCs and VSwitches exist in the zone where the RDS instance is located. For more information about how to create VPCs and VSwitches, see Manage VPCs.
Migrate the RDS instance from the classic network to a VPC

1. Log on to the ApsaraDB for RDS console.

2. In the upper-left corner of the page, select the region where the instance is located.

3. Find the instance and click the instance ID.

4. In the left-side navigation pane, click Database Connections.

5. Click Switch to VPC.

6. In the dialog box that appears, select a VPC and VSwitch, and select whether to retain the internal and public endpoints of the classic network.

   - Select a VPC. We recommend that you select the VPC where your ECS instance is located. Otherwise, the ECS instance and RDS instance cannot communicate through
the internal connections unless you create an express connection or gateway. For more information, see Alibaba Cloud CEN tutorials and VPN gateway.

- Select a VSwitch. If no VSwitch exists in the selected VPC (as shown in the following figure), create a VSwitch in the same zone as the instance. For more information, see Manage VSwitches.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear</td>
<td>The endpoint of the classic network is not retained. The original endpoint is changed to the endpoint of the VPC. If the endpoint of the classic network is not retained, a 30-second transient disconnection may occur to the RDS instance when the network type is changed. The internal access to the RDS instance from the ECS instance that is located in the classic network will be immediately disconnected.</td>
</tr>
<tr>
<td>Action</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Select</td>
<td>The endpoint of the classic network is retained, and a new endpoint of the VPC is added. Indicates that the hybrid access mode is used and RDS can be simultaneously accessed by ECS instances both in the classic network and VPC through the internal endpoints. If the endpoint of the classic network is retained, the RDS instance will not be immediately disconnected when the network type is changed. The ECS instances in the classic network will not be disconnected from the internal access to the RDS instance until the internal endpoint of the classic network expires. Before the endpoint of the classic network expires, add the endpoint of the VPC to the ECS instance that is located in the same VPC. This makes sure that your business is smoothly migrated to the VPC. Within seven days before the endpoints of the classic network expire, the system will send a text message to the mobile phone bound to your account every day.</td>
</tr>
</tbody>
</table>

**Database Connection**

- **Network Type:** VPC (VPC: `...`)
- **Internal Endpoint:** `...`.`mysql.rds.aliyuncs.com`

**Note:** Use the preceding connection string to connect to the instance. You need to change the VIP in the connection string.

**Original classic endpoint (Expired and released in 14 days)**

- **Intranet Address (Classic Network):** `...`.`mysql.rds.aliyuncs.com`
7. Add the internal IP address of the ECS instance in the VPC to the **VPC whitelist group** of the RDS instance. This makes sure that the ECS instance can access the RDS instance through the internal network. If no VPC whitelist group exists, create a new group.

![VPC whitelist group](image)

8. • If you select Retain Classic Network, add the endpoint of the VPC to the ECS instance before the endpoint of the classic network expires.

• If you clear Retain Classic Network, the internal connection from the ECS instance in the VPC to the RDS instance is immediately disconnected after the network type is changed. You must add the RDS endpoint of the VPC to the ECS instance.

**Note:**

To connect an ECS instance in the classic network to an RDS instance in a VPC through the internal network, you can use ClassicLink or switch the network type to VPC.

**Change the expiration time for the original internal endpoint of the classic network**

During the hybrid access period, you can change the retention period for the original internal endpoint of the classic network at any time as needed. The system updates the expiration date based on the modified date. For example, if the original internal endpoint of the classic network is set to expire on August 18, 2017, and you change the expiration time to "14 days later" on August 15, 2017. The internal endpoint of the classic network is released on August 29, 2017.

Follow these steps to change the expiration time:

1. Log on to the **ApsaraDB for RDS console**.
2. In the upper-left corner of the page, select the region where the instance is located.

3. Find the instance and click the instance ID.

4. In the left-side navigation pane, click **Database Connections**.

5. On the **Instance Connection** tab, click **Change Expiration Time**, as shown in the following figure.

6. On the **Change Expiration Time** page that appears, select an expiration time and click **OK**.
19 Database proxy

19.1 Dedicated proxy

This topic introduces the dedicated proxy feature provided by ApsaraDB RDS for MySQL. This feature allows you to use advanced features such as read/write splitting, connection pool, and transaction splitting.

Prerequisites

- Your RDS instance runs one of the following MySQL versions and RDS editions:
  - MySQL 8.0 Enterprise Edition (kernel version of 20191204 or later)
  - MySQL 8.0 High-availability Edition (kernel version of 20190915 or later)
  - MySQL 5.7 Enterprise Edition (kernel version of 20191128 or later)
  - MySQL 5.7 High-availability Edition (kernel version of 20190925 or later)
  - MySQL 5.6 High-availability Edition (kernel version of 20200229 or later)

**Note:**
To view the kernel version, perform the following steps: Log on to the ApsaraDB for RDS console, find the target RDS instance, and navigate to the Basic Information page. Then, in the Configuration Information section, check whether the Upgrade Minor Version button exists. If the button exists, click it to view the kernel version. If the button does not exist, it indicates that you are already using the latest kernel version. For more information, see Update the kernel version of an ApsaraDB RDS for MySQL instance.

- The instance is not in Hangzhou Zone D or China South 1 Finance Zone B.

**Note:**
To enable database proxy for an instance in the preceding zones, you can migrate the instance to another zone and then enable the dedicated proxy feature. For more information, see Migrate an ApsaraDB RDS MySQL instance across zones.
A dedicated proxy uses dedicated computing resources. It has the following benefits:

- A proxy endpoint is used to connect to your database system. This reduces maintenance costs by relieving you from updating the endpoints on your application. The proxy endpoint remains valid unless you release the proxy instances in the dedicated proxy. For example, you may enable read/write splitting during peak hours and then disable read/write splitting and release read-only instances after peak hours. In such cases, you do not need to update the endpoints on your application because the proxy endpoint is still connected.

- A dedicated proxy serves your database system exclusively to ensure service stability and guarantee that you do not need to compete with other users for resources.

- You can scale out compute resources for a dedicated proxy to handle more workloads.

Billing

Starting from January 15, 2020, you are issued one proxy instance free of charge. If you want to add more proxy instances, you must pay additional fees. The following table lists the prices.

<table>
<thead>
<tr>
<th>Region</th>
<th>Code</th>
<th>Dedicated proxy</th>
</tr>
</thead>
<tbody>
<tr>
<td>China (Hangzhou)</td>
<td>cn-hangzhou</td>
<td>0.173</td>
</tr>
<tr>
<td>China (Shanghai)</td>
<td>cn-shanghai</td>
<td>0.173</td>
</tr>
<tr>
<td>China (Qingdao)</td>
<td>cn-qingdao</td>
<td>0.173</td>
</tr>
<tr>
<td>China (Beijing)</td>
<td>cn-beijing</td>
<td>0.173</td>
</tr>
<tr>
<td>China (Zhangjiakou-Beijing Winter Olympic)</td>
<td>cn-zhangjiakou</td>
<td>0.120</td>
</tr>
<tr>
<td>China (Hohhot)</td>
<td>cn-huhehaote</td>
<td>0.173</td>
</tr>
<tr>
<td>China (Shenzhen)</td>
<td>cn-shenzhen</td>
<td>0.173</td>
</tr>
<tr>
<td>China (Heyuan)</td>
<td>cn-heyuan</td>
<td>0.173</td>
</tr>
<tr>
<td>China (Chengdu)</td>
<td>cn-chengdu</td>
<td>0.173</td>
</tr>
</tbody>
</table>
### Region

<table>
<thead>
<tr>
<th>Region</th>
<th>Code</th>
<th>USD/hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>China (Hong Kong)</td>
<td>cn-hongkong</td>
<td>0.297</td>
</tr>
<tr>
<td>Japan (Tokyo)</td>
<td>ap-northeast-1</td>
<td>0.288</td>
</tr>
<tr>
<td>India (Mumbai)</td>
<td>ap-south-1</td>
<td>0.231</td>
</tr>
<tr>
<td>Singapore</td>
<td>ap-southeast-1</td>
<td>0.271</td>
</tr>
<tr>
<td>Australia (Sydney)</td>
<td>ap-southeast-2</td>
<td>0.273</td>
</tr>
<tr>
<td>Malaysia (Kuala Lumpur)</td>
<td>ap-southeast-3</td>
<td>0.253</td>
</tr>
<tr>
<td>Indonesia (Jakarta)</td>
<td>ap-southeast-5</td>
<td>0.271</td>
</tr>
<tr>
<td>Germany (Frankfurt)</td>
<td>eu-central-1</td>
<td>0.243</td>
</tr>
<tr>
<td>UK (London)</td>
<td>eu-west-1</td>
<td>0.280</td>
</tr>
<tr>
<td>UAE (Dubai)</td>
<td>me-east-1</td>
<td>0.377</td>
</tr>
<tr>
<td>US (Virginia)</td>
<td>us-east-1</td>
<td>0.237</td>
</tr>
<tr>
<td>US (Silicon Valley)</td>
<td>us-west-1</td>
<td>0.284</td>
</tr>
</tbody>
</table>

### Limits

- Dedicated proxy instances do not support SSL encryption.
- Dedicated proxy instances do not support compression protocols.
- Dedicated proxy instances do not support changing VSwitches.

### Precautions

- A transient disconnection may occur while you change the specifications of the primary instance or its read-only instances.
- If you use a proxy endpoint to connect your applications to your database system, but you have not enabled the transaction splitting feature, all transaction requests are routed to the primary instance.
• If a proxy endpoint is used for read/write splitting, the read consistency of non-transaction requests cannot be guaranteed. To ensure read consistency, encapsulate requests in transactions.
• The `SHOW PROCESSLIST` statement returns a combination of results from the primary instance and its read-only instances.
• If you execute multiple statements or stored procedures, read/write splitting is disabled and all subsequent requests sent over the current connection are routed to the primary instance. To enable read/write splitting again, you must terminate the current connection and establish a new one.
• The `/*FORCE_MASTER*/` and `/*FORCE_SLAVE*/` hints are supported. However, requests that contain hints have higher route priorities. These requests are not constrained by consistency or transaction limits. You must check whether these hints are suitable for your business before you use them. A hint must not contain statements that change environment variables such as `/*FORCE_SLAVE*/ set names utf8;`. Otherwise, an error may occur in the subsequent procedure.
• A connection is replicated to the primary instance and all read-only instances after you enable the dedicated proxy feature (connection mode: 1:N). We recommend that you use the same connection specifications for your primary and read-only instances. Otherwise, the number of service connections is the number of connections allowed by the minimum instance specifications.
• After you enable the dedicated proxy feature and create or restart a read-only instance, only new connection requests are routed to the read-only instance. Connection requests generated before instance creation or restart are not routed to the read-only instance.
• The `max_prepared_stmt_count` parameters of the primary and read-only instances must be the same.

Enable the dedicated proxy feature

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the instance is located.

3. Find the instance and click its ID.

4. In the left-side navigation pane, click **Database Proxy**.

5. Click **Enable now**.

6. Specify **Network Type** and **Proxy Instances**. Click **Enable**.

**Note:**
- You cannot set the network type of an instance based on standard SSDs or enhanced SSDs to Internet Address.
We recommend that you specify the number of proxy instances as the rounded-up integer of the total number of CPU cores of primary and read-only instances divided by 8. You can set a maximum of 60 proxy instances.

For example, if your primary instance has eight CPU cores and its read-only instance has four CPU cores, the recommended number of proxy instances is calculated as follows: \((8 + 4)/8 = 1.5\) (rounded up to 2).

### Overview of the Database Proxy page

After the dedicated proxy feature is enabled, you can use a proxy endpoint for read/write splitting, connection pool, and transaction splitting.
<table>
<thead>
<tr>
<th>Section</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy Endpoint</td>
<td>Instance ID</td>
<td>The ID of the proxy instance.</td>
</tr>
<tr>
<td></td>
<td>Instances Associated with Proxy</td>
<td>The number of proxy instances associated with the proxy endpoint. You can add proxy instances to enable the dedicated proxy to process more requests. After the public preview phase ends, you must pay for added proxy instances.</td>
</tr>
<tr>
<td></td>
<td>Read/Write Splitting</td>
<td>Specifies whether to enable read/write splitting for the proxy endpoint. For more information, see Read/write splitting.</td>
</tr>
<tr>
<td></td>
<td>Connection Pool</td>
<td>Specifies the connection pool type of the proxy endpoint. Valid values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enable Transaction Connection Pool (default): The transaction connection pool is applicable to scenarios with tens of thousands of connections.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enable Session Connection Pool: The session connection pool is applicable to scenarios with PHP short-lived connections.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Disable: The connection pool feature is disabled.</td>
</tr>
<tr>
<td></td>
<td>Transaction Splitting</td>
<td>Specifies whether to enable transaction splitting for the proxy endpoint. For more information, see Transaction splitting.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> You can click Enable or Disable to the right of Transaction Splitting to enable or disable this feature.</td>
</tr>
<tr>
<td>Section</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Endpoint</td>
<td></td>
<td>The proxy endpoint to which read/write splitting is bound.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
<td>You can click <strong>Change Endpoint</strong> to the right of the endpoint to modify</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the proxy endpoint in the following format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The endpoint must start with a lowercase letter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The endpoint can contain letters, digits, and hyphens (-).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The endpoint must be 1 to 40 characters in length.</td>
</tr>
<tr>
<td>Port</td>
<td></td>
<td>The port of the proxy endpoint.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
<td>You can click <strong>Change Port</strong> to the right of the port to modify the port.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid values: 1000 to 5999.</td>
</tr>
<tr>
<td>Endpoint Type</td>
<td></td>
<td>The network type of the proxy endpoint. You cannot change the network type.</td>
</tr>
<tr>
<td>Proxy Instance</td>
<td>Proxy Type</td>
<td>Only <strong>Dedicated Proxy</strong> is supported.</td>
</tr>
<tr>
<td>CPU and Memory</td>
<td></td>
<td>The CPU and memory of proxy instances. Only 2 Cores, 4 GB is supported.</td>
</tr>
<tr>
<td>Instances</td>
<td>Associated</td>
<td>The total number of proxy instances. You can set a maximum of 60 proxy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>instances.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
<td>We recommend that you specify the number of proxy instances as the rounded-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>up integer of the total number of CPU cores of primary and read-only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>instances divided by 8.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For example, if your primary instance has eight CPU cores and its read-only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>instance has four CPU cores , the recommended number of proxy instances is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>calculated as follows: (8 + 4)/8 = 1.5 (rounded up to 2).</td>
</tr>
</tbody>
</table>

Issue: 20200702
Adjust the number of proxy instances

**Note:**
A transient disconnection may occur while you adjust the number of proxy instances. Make sure that your application is configured with automatic reconnection policies.

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the instance is located.
3. Find the instance and click its ID.
4. In the left-side navigation pane, click Database Proxy.
5. In the Proxy Instance section of the Proxy Service tab, change the number in the Adjusted Instances column and click Apply in the Adjustment Plan column.
6. Set **Applied At** and click **OK**.

![Configure Proxy Resources](image)

**Note:** Configuring proxy resources may cause applications to disconnect. Make sure that your applications are configured with automatic reconnection policies.

**View monitoring data**

1. Log on to the ApsaraDB for RDS console.

2. In the upper-left corner of the page, select the region where the instance is located.

3. Find the instance and click its ID.

4. In the left-side navigation pane, click **Database Proxy**.

5. Click the **Monitoring Data** tab.

6. Select a time range and view the **CPU Utilization (%)** metric within that time range.

**Note:**
The CPU Utilization (%) metric indicates the CPU utilization of the physical host where proxy instances reside.

Disable the dedicated proxy feature

1. Log on to the ApsaraDB for RDS console.

2. In the upper-left corner of the page, select the region where the instance is located.

3. Find the instance and click its ID.

4. In the left-side navigation pane, click Database Proxy.

5. In the upper-right corner of the page, click Disable Proxy Service.

6. Click OK.
FAQ

• Why is the dedicated proxy feature not available even though my RDS instance runs MySQL 5.7 and the latest kernel version?

After you update the kernel version of your RDS instance that runs MySQL 5.7, you must disable the original read/write splitting feature before you can use the dedicated proxy feature.

• Which endpoint do I need to use for read/write splitting, the proxy endpoint or the read/write splitting endpoint?

The proxy endpoint and the read/write splitting endpoint are the same. The backend uses the proxy endpoint to implement read/write splitting.

Related operations

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_315</td>
<td>Modifies the dedicated proxy settings of an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_316</td>
<td>Enables or disables the dedicated proxy feature of an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_317</td>
<td>Queries details about the dedicated proxy of an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_318</td>
<td>Queries the endpoint of the dedicated proxy for an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_319</td>
<td>Modifies the endpoint of the dedicated proxy for an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_320</td>
<td>Queries the performance of the dedicated proxy for an ApsaraDB for RDS instance.</td>
</tr>
</tbody>
</table>

19.2 Read/write splitting

This topic describes the read/write splitting function of ApsaraDB RDS for MySQL in its dedicated proxy feature and how to enable this function.

If your primary instance needs to process a large number of read requests but only a small number of write requests, you can create one or more read-only instances to offload read requests from your primary instance. This allows you to ensure service stability. For more information, see Overview of ApsaraDB RDS for MySQL read-only instances.
After you create read-only instances, you can enable read/write splitting. In this case, the endpoint of the dedicated proxy that serves your database system is used for read/write splitting. After you add the endpoint of the dedicated proxy to your application, write requests are routed to the primary instance and read requests are routed to the read-only instances.

**Differences between the read/write splitting endpoint and the internal and public endpoints**

After you enable read/write splitting and add the endpoint of the dedicated proxy to your application, all requests are routed to this endpoint, and then to the primary and read-only instances based on the request types and the read weights of these instances.

If the internal or public endpoint of the primary instance is added to your application, all requests are routed to the primary instance. In this case, you must add the endpoints and read weights of the primary and read-only instances to your application to implement read/write splitting.

**Logic to route requests**

- The following requests are routed only to the primary instance:
  - Data manipulation language (DML) statements, which are INSERT, UPDATE, DELETE, and SELECT FOR UPDATE
  - All data definition language (DDL) statements used to perform operations such as creating databases or tables, deleting databases or tables, and changing schemas or permissions
  - All requests that are encapsulated in transactions
  - Requests for user-defined functions
  - Requests for stored procedures
  - Requests for EXECUTE statements
  - Requests for multi-statements
  - Requests that involve temporary tables
  - Requests for SELECT last_insert_id() statements
  - All requests to query or modify user environment variables
  - Requests for SHOW PROCESSLIST statements
  - All requests for KILL statements in SQL (Note that these are not KILL commands in Linux.)
• The following requests are routed to the primary instance or its read-only instances:
  - Read requests that are not encapsulated in transactions
  - Requests for COM_STMT_EXECUTE statements
• The following requests are routed to all of the primary and read-only instances:
  - All requests to modify system environment variables
  - Requests for USE statements
  - Requests for COM_STMT_PREPARE statements
  - Requests for COM_CHANGE_USER, COM_QUIT, and COM_SET_OPTION statements

Benefits

• Easier maintenance with a unified endpoint
  
  If you do not enable read/write splitting, you must add the endpoints of the primary and read-only instances to your application. This makes write requests routed to the primary instance and read requests routed to the read-only instances.

  If you enable read/write splitting, the endpoint of the dedicated proxy is used for read/write splitting. After your application is connected to this endpoint, requests are routed to the primary and read-only instances based on the read weights of these instances. This reduces maintenance costs.

  In addition, you can scale up the read capability of your database system by creating read-only instances. This relieves you from the need to modify the configuration data on your application.

• Higher performance and lower maintenance cost with a native RDS link
  
  If you build a separate proxy layer on the cloud to implement read/write splitting, statements need to be parsed and forwarded by a number of components before they reach your database system. This increases response latency. The read/write splitting function provided with ApsaraDB for RDS shortens response latency, increases processing speed, and reduces maintenance costs.

• Ideal in various use scenarios with configurable read weights and thresholds
  
  You can specify the read weights of the primary and read-only instances. You can also specify a latency threshold for each read-only instance.

• Highly available with instance-level health check
  
  The read/write splitting module actively performs health checks on the primary and read-only instances. If an instance breaks down or its latency exceeds the specified threshold
the read/write splitting module stops routing requests to the instance and redirects the requests that were destined for the instance to other healthy instances. This allows you to ensure service availability in the event of faults on a single instance. After the instance is recovered, the read/write splitting module resumes routing read requests to it.

**Note:**
To avoid single points of failure (SPOFs), we recommend that you create at least two read-only instances.

**Precautions**

- A brief disconnection may occur while you change the specifications of the primary instance or its read-only instances.
- After you create a read-only instance, only the requests over new connections can be routed to the read-only instance.
- The endpoint of the dedicated proxy does not support SSL encryption.
- The endpoint of the dedicated proxy does not support compression.
- If the endpoint of the dedicated proxy is used for connection, all of the requests encapsulated in transactions are routed to the primary instance.
- If the endpoint of the dedicated proxy is used for read/write splitting, the read consistency of the requests that are not encapsulated in transactions cannot be ensured. If you require the read consistency, encapsulate the requests in transactions.
- If the endpoint of the dedicated proxy is used for connection, the `SHOW PROCESSLIST` statement combines the results from the primary and read-only instances and returns a result set.
- If short-lived connection optimization is enabled, the `SHOW PROCESSLIST` statement may return idle connections.
- If you execute multi-statements or stored procedures, read/write splitting is disabled and all of the subsequent requests over the current connection are routed to the primary instance. To enable read/write splitting again, you must terminate the current connection and establish a new one.
- The /*FORCE_MASTER*/ and /*FORCE_SLAVE*/ hints are supported. However, requests that contain hints have higher route priorities. These requests are not constrained by consistency or transaction limits. You must check whether these hints are suitable for your business before you use them. A hint cannot contain statements that change
environment variables. An example is "FORCE_SLAVE" = set names utf8; . Otherwise, an error may occur in the subsequent procedure.

Prerequisites

Before you enable read/write splitting for an RDS instance, make sure that the following requirements are met:

- The RDS instance is a primary instance. You cannot enable read/write splitting for a read-only or disaster recovery instance.
- The dedicated proxy feature is enabled for the RDS instance.
- The RDS instance is attached with read-only instances.

Enable read/write splitting

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click Database Proxy.
5. On the Read/Write Splitting tab, click Enable now.
6. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency Threshold</td>
<td>The maximum latency that is allowed for data replication from the primary instance to its read-only instances. If the replication latency for a read-only instance exceeds the specified threshold, the read/write splitting module stops routing read requests to the read-only instance. This applies even if the read-only instance has a high read weight. Valid values: 0 to 7200. Unit: seconds. The read-only instances may replicate data from the primary instance at a certain latency due to SQL statement execution limits. We recommend that you set this parameter to a value that is greater than or equal to 30.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Read Weight Distribution</strong></td>
<td>The read weight of each instance in your database system. A higher read weight indicates more read requests to process. For example, the primary instance is attached with three read-only instances, and the read weights of the primary and read-only instances are 0, 100, 200, and 200, respectively. In this situation, the primary instance only processes write requests, and the three read-only instances process all of the read requests at the 1:2:2 ratio.</td>
</tr>
<tr>
<td><strong>• Automatic Distribution</strong>: Your database system assigns a read weight to each instance based on the instance specifications. After you create a read-only instance, your database system assigns a read weight to the read-only instance and adds the read-only instance to the read/write splitting link. For more information, see Rules of weight allocation by the system.</td>
<td></td>
</tr>
<tr>
<td><strong>• Customized Distribution</strong>: You must manually specify the read weight of each instance. Valid values: 0 to 10000. After you create a read-only instance, its read weight defaults to 0. You must manually specify the read weight of the read-only instance.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
You cannot specify the read weight of a read-only instance for which you have specified a replication latency. For more information, see Set a replication delay for an RDS MySQL read-only instance.
7. Click OK.

After opening, you need to configure the read/write splitting endpoint (proxy endpoint) in the application program to automatically forward the write request to the primary instance and the read request to each read-only instance.

FAQ

Does read/write splitting support hints?

Yes, you can use hints to specify to forward requests to your primary instance. For more information about the hint formats that are supported by ApsaraDB for RDS, see Rules of weight allocation by the system.

References

FAQ for read/write splitting

19.3 Connection pool

ApsaraDB for RDS provides connection pools for dedicated proxy instances. You can select a connection pool type based on your business needs. This reduces workloads on your
RDS instances caused by excessive connections or frequent short-lived connections (for example, when PHP is used).

Prerequisites

The dedicated proxy is enabled for your RDS instance. For more information, see Dedicated proxy.

Context

The dedicated proxy supports the following two types of connection pools:

- **Transaction connection pool** (default)

  The dedicated proxy uses a transaction connection pool by default. You can use this type of connection pool in scenarios where tens of thousands of connections are established.

  The transaction connection pool can be used to reduce the number of connections to the database and workloads caused by frequent short-lived connections. Database clients can establish a large number of connections with the proxy, and the proxy only creates a small number of connections with the database. When a client sends a request to the database, the proxy establishes a connection that matches the system variables specified in the request with the database. After the transaction is complete, the connection is released to the connection pool.

  For information about limits on the transaction connection pool, see Limits on the transaction connection pool.

- **Session connection pool**

  The session connection pool applies to scenarios where short-lived connections are frequently established.

  It can be used to reduce workloads on the RDS instance caused by frequent short-lived connections. When a client is disconnected, the proxy checks whether the current connection is idle. If it is idle, the proxy retains the connection in the connection pool for a short period of time. When the client initiates a connection request again, the proxy matches the request with idle connections retained in the connection pool based on the values of the user, clientip, and dbname fields. If an idle connection in the connection pool is matched, the proxy uses this idle connection. If no idle connection is matched
, a new connection is established with the database. This reduces the overheads of database connections.

**Note:**
The session connection pool does not reduce concurrent connections with the database. It reduces the frequency to establish connections between the application and database and workloads of the primary MySQL thread. This improves efficiency to process business requests. However, idle connections in the connection pool still occupy the database threads for a short period of time.

**Precautions**

You cannot configure different permissions for the same account with different source IP addresses. Otherwise, errors may occur when connections in the connection pool are reused. For example, if the user account has permissions on database_a when its source IP address is 192.168.1.1 but does not have permissions on database_a when its source IP address is 192.168.1.2, the connection pool may encounter permission errors.

**Limits on the transaction connection pool**

- When the following operations are performed, the proxy locks the connections and does not release them to the connection pool until the transactions are complete.
  - Execute the PREPARE statement.
  - Create a temporary table.
  - Modify user variables.
  - Process large packets, for example, packets larger than 16 MB.
  - Execute the LOCK TABLE statement.
  - Execute multiple statements.
  - Call a stored procedure.
- The FOUND_ROWS, ROW_COUNT, and LAST_INSERT_ID functions are not supported. These functions can be called but may return incorrect results.
- When a client is connected to the proxy, the proxy obtains a connection to the database from the connection pool. If the connection has the wait_timeout parameter configured, the database terminates the connection with the proxy after the time specified by wait_timeout elapses. However, the client may still be connected with the proxy.
- The connection pool matches requests with connections based on the sql_mode, character_set_server, collation_server, and time_zone variables. If the requests include
other session-level system variables, after the connections are established, you must explicitly execute the SET statement to set these additional variables. Otherwise, the connection pool may reuse connections whose system variables are changed.

- You can use the SELECT CONNECTION_ID() statement to query the thread ID of the current connection to determine whether the connection is reused.
- The IP addresses and ports returned by the SHOW PROCESSLIST statement may be different from the actual IP addresses and ports of clients because connections may be reused.
- The database proxy merges the results of the SHOW PROCESSLIST statement for all RDS instances and returns the results to clients. After you enable the transaction connection pool, the thread ID of the connection between the client and the proxy is not consistent with that between the proxy and the database. In this case, when you kill a process, the kill command may return an error message even if it is successfully executed. You can execute the SHOW PROCESSLIST statement again to check whether the process is killed.

**Change the connection pool**

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target instance and click its ID.
4. In the left-side navigation pane, click **Database Proxy**.
5. On the right of **Connection Pool**, select a connection pool type.
The change of the connection pool immediately takes effect.

### 19.4 Transaction splitting

This topic introduces the transaction splitting function provided by ApsaraDB RDS for MySQL in its dedicated proxy feature. This function reduces the loads on the primary instance by identifying and distributing the read requests initiated prior to write requests within a transaction to read-only instances.

**Prerequisites**

For more information, see [Dedicated proxy](#).

**Context**

By default, the dedicated proxy sends all requests encapsulated in transactions to the primary instance to ensure the correctness of the transactions. If the framework used encapsulates all requests in transactions, the primary instance becomes heavily loaded. In such cases, you can enable the transaction splitting function.

In the default transaction isolation level READ COMMITTED, after ApsaraDB RDS MySQL turns off automatic transaction commit (set autocommit = 0), Read requests initiated before the transaction is started are distributed to read-only instances by using the SLB module.

**Note:**

- Explicit transactions (for example, begin or start transaction) do not support transaction splitting.
If you enable transaction splitting, global consistency cannot be ensured for some services. We recommend that you evaluate whether transaction splitting is suitable for your business before you enable it.

**Procedure**

1. Log on to the ApsaraDB for RDS console.

2. In the upper-left corner of the page, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID.

4. In the left-side navigation pane, click **Database Proxy**.
5. In the Proxy Endpoint section of the Proxy Service tab, click **Enable** to the right of **Transaction Splitting**.

**Note:**

- When you no longer need transaction splitting, you can click **Disable** to the right of **Transaction Splitting** to disable it.
- After you enable or disable transaction splitting, the setting only works for new connections.

### 19.5 Read/write splitting

#### 19.5.1 Enable the read/write splitting function in the shared proxy of an ApsaraDB RDS for MySQL instance

This topic describes how to enable the read/write splitting function in the shared proxy of an ApsaraDB RDS for MySQL instance. This function allows your application to connect to your database system by using a read/write splitting endpoint. Your database system distributes write requests to the primary instance and read requests to the read-only instances attached to the primary instance based on the specified read weights.

**Note:**

You can promote the proxy of the primary instance from shared to dedicated. A dedicated proxy performs better than a shared proxy. We recommend that you enable read/write splitting after you promote the proxy of the primary instance from shared to dedicated. For more information, see **Introduction to read/write splitting in the dedicated proxy.**

**Prerequisites**

Before you enable read/write splitting for an RDS instance, make sure that the following requirements are met:
• The RDS instance is a primary instance.
• The RDS instance is running one of the following MySQL versions and RDS editions:
  - MySQL 5.7 in the High-availability Edition (with local SSDs)
  - MySQL 5.6
• The RDS instance is attached with at least one read-only instance. For more information, see Create an ApsaraDB RDS for MySQL read-only instance.
• No replication latencies are configured for the read-only instances that are attached to the primary instance. Otherwise, read/write splitting cannot take effect. For more information, see Set a replication delay for an RDS MySQL read-only instance.

**Billing**

The read/write splitting function is free of charge. However, you still need to pay for the read-only instances that you create.

**Limits (in shared proxy)**

• If the length of an SQL statement exceeds 64 KB, the read/write splitting link does not parse the SQL statement and directly routes it to the primary instance. This applies even if you use a hint to specify to send the SQL statement to the primary instance. If you want to execute an SQL statement to query data from a read-only instance, make sure that the length of the SQL statement does not exceed 64 KB.
Statements that are prepared by using the PREPARE statement in the SQL command line tool are executed on the primary instance.

Statements that are prepared by using the PREPARE statement in the MySQL command line tool are forwarded to the read-only instances until the PREPARE statement is terminated.

The SET GLOBAL, SET USER, and SET ONCE statements that are used to configure environment variables are executed on the primary instance.

The following statements and functions are not supported:

- SSL encryption
- Compression
- com_dump_table and com_change_user
- KILL CONNECTION [QUERY]
- CHANGE USER

The execution result is random for the following statements:

The SHOW PROCESSLIST, SHOW MASTER STATUS, and COM_PROCESS_INFO statements return results based on the instance to which they are routed.

All requests in transactions are routed to the primary instance.

**Note:**

If an SQL statement in a transaction fails to be executed, the transaction may be rolled back. If the transaction is rolled back and the subsequent SELECT statement in it is routed to a read-only instance, the query result may be abnormal. We recommend that you resolve the issue in the failed SQL statement before you continue to execute the transaction.

Read/write splitting does not ensure the read consistency of requests that are not encapsulated in transactions. If you require the read consistency, add hints that specify to route the requests to the primary instance or encapsulate the requests in transactions.

The LAST_INSERT_ID() function is not supported. If you want to use this function in a request, add the following hint to the request: hint: /*FORCE_MASTER*/, eg:/*FORCE_MASTER*/ SELECT LAST_INSERT_ID();

**Precautions**

- If it is the first time that you enable the read/write splitting function, your database system automatically upgrades the backend administration systems of the primary and
read-only instances to the latest version. This ensures service availability. While the read/write splitting function is being enabled, the primary instance is disconnected for 30 seconds or less. All of the read-only instances are inaccessible during the brief disconnection. We recommend that you enable the read/write splitting function during off-peak hours and make sure that your application is configured to automatically reconnect to the primary instance.

- If you have restarted or changed the specifications of the primary and read-only instances at least once after March 8, 2017, your database system has upgraded. This includes the upgrade of the backend administration systems of these instances to the latest version. In this situation, when you enable the read/write splitting function, your database system does not restart these instances and no brief disconnection occurs.

- The endpoint that is generated for read/write splitting is fixed. If you enable and disable the read/write splitting function multiple times, the endpoint remains unchanged. You do not need to perform a regular update of the configuration data on your application. This reduces maintenance costs.

**Note:**

The read/write splitting endpoint cannot be manually changed.

**Procedure**

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target primary instance resides.

3. Find the target primary instance and click its ID.

4. In the left-side navigation pane, click **Database Connection** or **Database Proxy**.

5. On the **Read/Write Splitting** tab, click **Enable now**.

**Note:**

- If the Enable now button does not appear, you must enable the database proxy feature and create at least one read-only instance.
- When you enable the read/write splitting function, it requires a restart of the primary and read-only instances. This applies when the instances were created before March 8, 2017 and you have not restarted them or changed their specifications. In such cases, you must click **OK** in the displayed dialog box to enable the read/write splitting function.
6. Configure the following parameters.

**Configure Read/Write Splitting**

<table>
<thead>
<tr>
<th>Network Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intranet address (Classic network)</td>
</tr>
<tr>
<td>Internet Address</td>
</tr>
</tbody>
</table>

**Latency Threshold:**

30 Second

The read requests are not distributed to the read-only instance whose latency exceeds the threshold.

**Read Weight Distribution**

<table>
<thead>
<tr>
<th>Read Weight Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic Distribution</td>
</tr>
<tr>
<td>Customized Distribution</td>
</tr>
</tbody>
</table>

* How to set the weight?*

<table>
<thead>
<tr>
<th>Master instance</th>
<th>Read-only instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>rm-xxxxxx</td>
<td>0</td>
</tr>
<tr>
<td>rr-xxxxxx</td>
<td>100</td>
</tr>
</tbody>
</table>

* The system distributes the weight automatically. The weights of the subsequent new read-only instances will be automatically distributed according to the system weight distribution rules.
* The weight of the instance will be removed when the instance is in the downtime or when its delay times out. After the instance is restored, the weight will be automatically restored.
* The weight of the instance will be automatically removed after the instance is released.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Type</td>
<td>The type of the read/write splitting endpoint.</td>
</tr>
<tr>
<td></td>
<td>• Intranet Address:</td>
</tr>
<tr>
<td></td>
<td>- If the network type of the primary instance is VPC, the network type</td>
</tr>
<tr>
<td></td>
<td>of the read/write splitting endpoint is VPC.</td>
</tr>
<tr>
<td></td>
<td>- If the network type of the primary instance is classic network, the</td>
</tr>
<tr>
<td></td>
<td>network type of the read/write splitting endpoint is classic network.</td>
</tr>
<tr>
<td></td>
<td>• Internet Address: used to connect to the primary instance over the</td>
</tr>
<tr>
<td></td>
<td>Internet. The Internet may be prone to fluctuations in connectivity.</td>
</tr>
<tr>
<td></td>
<td>We recommend that you connect to the primary instance by using an</td>
</tr>
<tr>
<td></td>
<td>internal endpoint.</td>
</tr>
<tr>
<td>Latency Threshold</td>
<td>The maximum latency that is allowed for the data replication from the</td>
</tr>
<tr>
<td></td>
<td>primary instance to the read-only instances. If the replication latency</td>
</tr>
<tr>
<td></td>
<td>of a read-only instance exceeds this threshold, the read/write splitting</td>
</tr>
<tr>
<td></td>
<td>link no longer distributes read requests to the read-only instance. For</td>
</tr>
<tr>
<td></td>
<td>example, it is not based on a high read weight of the read-only instance.</td>
</tr>
<tr>
<td></td>
<td>Valid values: 0 to 7200. Unit: seconds. The read-only instances may</td>
</tr>
<tr>
<td></td>
<td>replicate data from the primary instance at a certain latency due to SQL</td>
</tr>
<tr>
<td></td>
<td>statement execution limits. We recommend that you set this parameter to</td>
</tr>
<tr>
<td></td>
<td>a value that is greater than or equal to 30.</td>
</tr>
</tbody>
</table>
### Parameter | Description
--- | ---
**Read Weight Distribution** | The read weight of each instance in your database system. A higher read weight indicates more read requests to process. For example, the primary instance is attached with three read-only instances, and the read weights of the primary and read-only instances are 0, 100, 200, and 200, respectively. In this situation, the primary instance only processes write requests, and the read-only instances process all of the read requests at the 1:2:2 ratio.

• **Automatic Distribution**: Your database system assigns a read weight to each instance based on the instance specifications. After you create a read-only instance, your database system assigns a read weight to the read-only instance and adds the read-only instance to the read/write splitting link. For more information, see [Rules of weight allocation by the system](#).

• **Customized Distribution**: You must manually specify the read weight of each instance. Valid values: 0 to 10000. After you create a read-only instance, the default value of the read weight is 0. You must manually specify the read weight of the read-only instance.

**Note:**
You cannot specify the read weight of a read-only instance for which you have specified a replication latency. For more information, see [Set a replication delay for an RDS MySQL read-only instance](#).

7. Click **OK**.

**Note:**
The status of the primary instance changes to **Creating Network Connection**. Wait until the status changes back to **Running**.

**What to do next**

You can modify the configuration data on your application, so your application connects to the primary instance by using the read/write splitting endpoint.

**Note:**
You can obtain the read/write splitting endpoint of the primary instance on the **Basic Information** page or on the Read/Write Splitting tab of the **Database Connection** page.

**FAQ**

- Modify the latency threshold and read weights
• Change the network type of the read/write splitting address for an RDS MySQL instance
• Monitor the read/write splitting performance of an RDS MySQL instance
• Rules of weight allocation by the system
• FAQ for read/write splitting

19.5.2 Change the network type of the read/write splitting address for an RDS MySQL instance

This topic describes how to change the network type of the read/write splitting address for an RDS MySQL instance. When you enable the read/write splitting function, you can use the internal or public endpoint of the master instance as the read/write splitting address.

If your application is deployed in the same VPC as the RDS instance, we recommend that you use the internal endpoint to guarantee data security and communication efficiency. If your application is deployed on your computer or in a different network from your RDS instance, you can use the public endpoint to establish a connection.

Prerequisites

• The read/write splitting function is enabled. For more information, see Enable the read/write splitting function in the shared proxy of an ApsaraDB RDS for MySQL instance.

• The MySQL version and RDS edition are as follows:
  - MySQL 5.7 High-availability Edition (with local SSDs)
  - MySQL 5.6

Note:
The read/write splitting function is available in MySQL 8.0 only when the dedicated database proxy service is enabled. However, MySQL 8.0 supports only the VPC network type, therefore you cannot switch to the public endpoint.
Precautions

When you change the network type of the read/write splitting address, the master instance experiences a transient disconnection for up to 30 seconds. To avoid the impact of the transient disconnection, we recommend that you change the network type during off-peak hours and make sure that your application can automatically reconnect to the RDS instance.

Procedure

1. Log on to the RDS console.
2. Select the target region.
3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click Database Connection or Database Proxy.

5. Click the Read/Write Splitting tab.

6. Click Switch to Internet Address or Switch to Inner Address.

7. In the displayed dialog box, click Confirm.

Reference

FAQ for read/write splitting

19.5.3 Disable read/write splitting for an RDS MySQL instance

This topic describes how to disable read/write splitting for an RDS MySQL instance.

Prerequisites

The read/write splitting function has been enabled for the RDS instance. For more information, see Enable the read/write splitting function in the shared proxy of an ApsaraDB RDS for MySQL instance.

Precautions

- When the read/write splitting function is being disabled, the RDS instance is disconnected for 30 seconds or less. We recommend that you disable this function during off-peak hours and make sure that your application can automatically reconnect to the RDS instance.
- After the read/write splitting function is disabled, the read/write splitting address becomes invalid. Make sure that your application no longer uses the read/write splitting address to connect to the RDS instance.

Procedure

1. Log on to the RDS console.
2. In the upper-left corner, select the region where the target RDS instance is located.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click Database Connection or Database Proxy.

5. Click the Read/Write Splitting tab.

6. In the Basic Information of Read/Write Splitting section, click Disable Read/Write Splitting.

7. In the displayed dialog box, click Confirm.

19.5.4 Monitor the read/write splitting performance of an RDS MySQL instance

This topic describes how to monitor the read/write splitting performance of an RDS MySQL instance in the RDS console.

Procedure

1. Log on to the RDS console.

2. In the upper-left corner, select the region where the target RDS instance is located.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click Monitoring and Alerts.
5. On the **Monitoring** tab, select **Engine Monitoring**.

![Engine Monitoring](image)

6. View the **TPS (Transactions per Second)/QPS (Queries per Second)** chart to obtain the number of read and write operations on each of the master and read-only instances.

**19.5.5 Verify the read/write splitting performance of an RDS MySQL instance**

This topic uses Sysbench 0.5, a MySQL stress testing tool, as an example to describe how to correctly configure parameters to verify the read/write splitting performance of an RDS MySQL instance. After read/write splitting is enabled, all transactions are routed to the master instance by default.

**Prerequisites**

- The read/write splitting function is enabled. For detailed operations, see [Enable the read/write splitting function in the shared proxy of an ApsaraDB RDS for MySQL instance](#).
- The Sysbench 0.5 is installed. For detailed operations, see the [Sysbench documentation](#).

**Precautions**

- We recommend that a case with prepare or a transaction not be for testing the load balance performance of read/write splitting.
- Prevent the master/slave latency from exceeding the threshold set for the monitoring check due to high read stress.
- We recommend that you use the following Sysbench scripts to build a specific SQL statement as needed.

```plaintext
function thread_init(thread_id)
    db_connect()
end

function event(thread_id)
    rs = db_query("select 1")
```

---

*Issue: 20200702*
Set Sysbench parameters

A transaction is used by default to test the Sysbench oltp.lua script. If you use default parameters, all SQL statements are executed in the transaction and read-only SQL statements are routed to the master database for execution. Therefore, when the Sysbench is used to test read/write splitting performance, you must set the Sysbench parameters as needed. For example, you can set the `oltp-skip-trx` parameter to make sure that the Sysbench does not run the SQL statement in a transaction.

Set common parameters

You can set the following parameters as needed.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>test</td>
<td>Path of the test file</td>
</tr>
<tr>
<td>mysql-host</td>
<td>IP address of the MySQL server</td>
</tr>
<tr>
<td>mysql-port</td>
<td>Port of the MySQL server</td>
</tr>
<tr>
<td>mysql-user</td>
<td>User name</td>
</tr>
<tr>
<td>mysql-password</td>
<td>Password</td>
</tr>
<tr>
<td>mysql-db</td>
<td>Database for testing, which must be created in advance</td>
</tr>
<tr>
<td>oltp-tables-count</td>
<td>Number of created tables</td>
</tr>
<tr>
<td>oltp-table-size</td>
<td>Number of records generated in each table</td>
</tr>
<tr>
<td>rand-init</td>
<td>Whether data is randomly initialized</td>
</tr>
<tr>
<td>max-time</td>
<td>Stress testing duration</td>
</tr>
<tr>
<td>max-requests</td>
<td>Total number of requests within a stress testing duration</td>
</tr>
<tr>
<td>num-threads</td>
<td>Number of concurrent threads</td>
</tr>
<tr>
<td>report-interval</td>
<td>Reporting interval of operating logs</td>
</tr>
</tbody>
</table>

Set parameters for transactions and read/write SQL statements

The following parameters can affect transactions and read/write SQL statements. Therefore, you must set parameters in read/write splitting tests as needed.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| oltp-test-mode | Indicates the test mode. This parameter is unavailable in Sysbench 0.5, so this parameter can be ignored. Possible values:  
• complex: Default value. For transactional tests.  
• simple: Simple test for read-only SQL statements.  
• nontrx: For non-transactional tests.  
• sp: Stored procedures. |
| oltp-skip-trx  | Indicates whether "begin" and "commit" of SQL statements are omitted. Possible values:  
• off: Default value. All SQL statements are executed in transactions.  
• on: Non-transactional mode. If a comparative stress test is executed repeatedly, you must run prepare and cleanup again. |
| oltp-read-only | Indicates whether read-only SQL statements are generated. Possible values:  
• off: Default value. The mixed read/write SQL statements of oltp.lua is executed.  
• on: Only read-only SQL statements are generated. UPDATE, DELETE, and INSERT SQL statements are not applicable. |

**Note:** When a stress test is executed to test the read/write splitting performance, you must set it to on and omit "begin" and "commit" of SQL statements.

**Note:** Set the parameter value as needed to perform read-only or read/write tests.

**Stress testing examples**

**Test read/write performance**
1. Run the following command to prepare data:

```bash
sysbench --test=./tests/db/oltp.lua --mysql-host=127.0.0.1 --mysql-port=3001 --
mysql-user=abc --mysql-password=abc123456 --mysql-db=testdb --oltp-tables-
count=10 --oltp-table-size=500000 --report-interval=5 --oltp-skip-trx=on --oltp-read
-only=off --rand-init=on --max-requests=0 --max-time=300 --num-threads=100
prepare;
```

2. Run the following command to conduct the test:

```bash
Note:
When data is updated for non-transactional read/write tests, errors such as ALERT:
Error 1062 Duplicate entry 'xxx' for key 'PRIMARY' may occur. You must add --mysql-
ignore-errors=1062 to skip these errors. If the parameter mysql-ignore-errors does not
take effect, your current Sysbench version is too old and you must upgrade it to the
latest version.

sysbench --test=./tests/db/oltp.lua --mysql-host=127.0.0.1 --mysql-port=3001 --
mysql-user=abc --mysql-password=abc123456 --mysql-db=testdb --oltp-tables-
count=10 --oltp-table-size=500000 --report-interval=5 --oltp-skip-trx=on --oltp-read
-only=off --mysql-ignore-errors=1062 --rand-init=on --max-requests=0 --max-time=
300 --num-threads=100 run;
```

3. Run the following command to clean up data:

```bash
sysbench --test=./tests/db/oltp.lua --mysql-host=127.0.0.1 --mysql-port=3001 --
mysql-user=abc --mysql-password=abc123456 --mysql-db=testdb --oltp-tables-
count=10 --oltp-table-size=500000 --report-interval=5 --oltp-skip-trx=on --oltp-read
-only=off --rand-init=on --max-requests=0 --max-time=300 --num-threads=100
cleanup;
```

**Test read-only performance**

1. Run the following command to prepare data:

```bash
sysbench --test=./tests/db/oltp.lua --mysql-host=127.0.0.1 --mysql-port=3001 --
mysql-user=abc --mysql-password=abc123456 --mysql-db=testdb --oltp-tables-
count=10 --oltp-table-size=500000 --report-interval=5 --oltp-skip-trx=on --oltp-read-only=on --
rand-init=on --max-requests=0 --max-time=300 --num-threads=100
prepare;
```

2. Run the following command to conduct the test:

```bash
sysbench --test=./tests/db/oltp.lua --mysql-host=127.0.0.1 --mysql-port=3001 --
mysql-user=abc --mysql-password=abc123456 --mysql-db=testdb --oltp-tables-
count=10 --oltp-table-size=500000 --report-interval=5 --oltp-skip-trx=on --oltp-read-only=on --
rand-init=on --max-requests=0 --max-time=300 --num-threads=100 run;
```

3. Run the following command to clean up data:

```bash
sysbench --test=./tests/db/oltp.lua --mysql-host=127.0.0.1 --mysql-port=3001 --
mysql-user=abc --mysql-password=abc123456 --mysql-db=testdb --oltp-tables-
count=10 --oltp-table-size=500000 --report-interval=5 --oltp-skip-trx=on --oltp-
19.5.6 Modify the latency threshold and read weights of ApsaraDB RDS for MySQL instances

This topic describes how to modify the latency threshold and read weights of a primary ApsaraDB RDS for MySQL instance and its read-only instances after you enable read/write splitting.

For more information, see the "Read/write splitting parameters" section.

Procedure

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click Database Connection or Database Proxy.
5. Click the Read/Write Splitting tab.
6. In the Basic Information of Read/Write Splitting section, click **Configure Read/Write Splitting** and configure parameters as prompted.

Configure Read/Write Splitting

Latency Threshold: 30 Seconds

The read requests are not distributed to the read-only instance whose latency exceeds the threshold.

Read Weight Distribution

<table>
<thead>
<tr>
<th></th>
<th>Master instance</th>
<th>Read-only instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>rm: -----</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>rr: -----</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* The system distributes the weight automatically. The weights of the subsequent new read-only instances will be automatically distributed according to the system weight distribution rules.

* The weight of the instance will be removed when the instance is in the downtime or when its delay times out. After the instance is restored, the weight will be automatically restored.

* The weight of the instance will be automatically removed after the instance is released.
### Table 19-1: Read/write splitting parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Latency Threshold</strong></td>
<td>The maximum latency that is allowed for data replication from the primary instance to its read-only instances. If the latency on a read-only instance exceeds the specified threshold, your database system stops routing read requests to the read-only instance. This applies even if the read-only instance has a high read weight. This mechanism prevents long-term data inconsistencies between the primary and read-only instances. Valid values: 0 to 7200. Unit: seconds. The read-only instances may replicate data from the primary instance at a certain latency due to SQL statement execution limits. We recommend that you set this parameter to a value that is greater than or equal to 30.</td>
</tr>
<tr>
<td><strong>Read Weight Distribution</strong></td>
<td>The read weight of each instance in your database system. A higher read weight indicates more read requests to process. For example, the primary instance is attached with three read-only instances, and the read weights of the primary and read-only instances are 0, 100, 200, and 0, respectively. In this situation, the primary instance only processes write requests, the first two read-only instances process all of the read requests at the 1:2 ratio, and the third read-only instance does not process write or read requests.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>• After you delete a read-only instance, its read weight is also removed, but the read weights of the other instances remain unchanged. • You cannot assign a read weight to a read-only instance for which you have specified a replication latency. For more information, see Set a replication delay for an RDS MySQL read-only instance.</td>
</tr>
</tbody>
</table>

7. Click **OK**.
(Optional) What to do next

- Verify the allocated read weights
- Use SQL Explorer to verify the effect of read/write splitting
- Monitor the read/write splitting performance of an RDS MySQL instance
- Verify the read/write splitting performance of an RDS MySQL instance

FAQ

- If the read weight of a read-only instance is 0, can I access the read-only instance by using the read/write splitting endpoint?
  
  No, you cannot access a read-only instance by using the read/write splitting endpoint if its read weight is 0. You can only access this read-only instance by using its internal or public endpoint. In most cases, this solution is used to make a read-only instance serve only a specific service.

- After I modify the read weights of my primary instance and its read-only instances, why cannot the new read weights take effect?
  
  The new read weights take effect only on new connections because the existing connections will not be terminated and re-established.

- The loads on my primary instance and its read-only instances do not comply with the specified read weights. Why?
  
  Possible reasons are as follows:
  
  - Requests contain transactions. All of the requests that contain transactions are routed to the primary instance. These transactions include those executed to read data.
  - Your database system is connected by using the endpoints of the primary and read-only instances. In such cases, your database system does not route requests to the primary and read-only instances based on the specified read weights. Make sure that your database system is connected only by using the read/write splitting endpoint.
• Why does a large number of read requests are routed to my primary instance even if the read weight of the primary instance is 0?

Possible reasons are as follows:

- The read requests contain transactions. All of the requests that contain transactions are routed to the primary instance. These transactions include those executed to read data.
- All of the read-only instances with a non-zero read weight are unavailable, or the latencies on these instances exceed the specified threshold. In this situation, your database system stops routing read requests to these instances.

References

FAQ for read/write splitting

Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_329</td>
<td>Modifies the latency threshold and read weights of a primary ApsaraDB for RDS instance and its read-only instances.</td>
</tr>
</tbody>
</table>

19.5.7 Rules of weight allocation by the system

This topic describes the default read weights allocated to RDS instances of various specifications.

Weight values list

When the read weights are automatically allocated for instances by the system, the values of these weights are fixed, as shown in the following table.

Table 19-2: Weights for RDS MySQL read-only instances

<table>
<thead>
<tr>
<th>Specification code</th>
<th>Specification type</th>
<th>Memory</th>
<th>CPU</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>rds.mysql.t1.small</td>
<td>General-purpose instance</td>
<td>1 GB</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>rds.mysql.s1.small</td>
<td>General-purpose instance</td>
<td>2 GB</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>Specification code</td>
<td>Specification type</td>
<td>Memory</td>
<td>CPU</td>
<td>Weight</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------</td>
<td>--------</td>
<td>-----</td>
<td>--------</td>
</tr>
<tr>
<td>rds.mysql.s2.large</td>
<td>General-purpose instance</td>
<td>4 GB</td>
<td>2</td>
<td>200</td>
</tr>
<tr>
<td>rds.mysql.s2.xlarge</td>
<td>General-purpose instance</td>
<td>8 GB</td>
<td>2</td>
<td>200</td>
</tr>
<tr>
<td>rds.mysql.s3.large</td>
<td>General-purpose instance</td>
<td>8 GB</td>
<td>4</td>
<td>400</td>
</tr>
<tr>
<td>rds.mysql.m1.medium</td>
<td>General-purpose instance</td>
<td>16 GB</td>
<td>4</td>
<td>400</td>
</tr>
<tr>
<td>rds.mysql.c1.large</td>
<td>General-purpose instance</td>
<td>16 GB</td>
<td>8</td>
<td>800</td>
</tr>
<tr>
<td>rds.mysql.c1.xlarge</td>
<td>General-purpose instance</td>
<td>32 GB</td>
<td>8</td>
<td>800</td>
</tr>
<tr>
<td>rds.mysql.c2.xlarge</td>
<td>General-purpose instance</td>
<td>64 GB</td>
<td>16</td>
<td>1600</td>
</tr>
<tr>
<td>rds.mysql.c2.xlp2</td>
<td>General-purpose instance</td>
<td>96 GB</td>
<td>16</td>
<td>1600</td>
</tr>
<tr>
<td>rds.mysql.c2.2xlarge</td>
<td>General-purpose instance</td>
<td>128 GB</td>
<td>16</td>
<td>1600</td>
</tr>
<tr>
<td>mysqlro.x8.medium.1</td>
<td>Dedicated instance</td>
<td>16 GB</td>
<td>2</td>
<td>200</td>
</tr>
<tr>
<td>mysqlro.x4.large.1</td>
<td>Dedicated instance</td>
<td>16 GB</td>
<td>4</td>
<td>200</td>
</tr>
<tr>
<td>mysqlro.x8.large.1</td>
<td>Dedicated instance</td>
<td>32 GB</td>
<td>4</td>
<td>400</td>
</tr>
<tr>
<td>mysqlro.x4.xlarge.1</td>
<td>Dedicated instance</td>
<td>32 GB</td>
<td>8</td>
<td>400</td>
</tr>
</tbody>
</table>
### Specify whether a SQL statement is sent to the master instance or a read-only instance by adding a hint

In addition to the weight distribution system of read/write splitting, hints serve as a complementary SQL syntax to specify whether a SQL statement is executed on the master instance or a read-only instance.

Hints supported by RDS read/write splitting are as follows:

- /*FORCE_MASTER*/: specifies that a SQL statement is executed on the master instance.
- /*FORCE_SLAVE*/: specifies that a SQL statement is executed on a read-only instance.

**Note:**
When you use the /*FORCE_SLAVE*/ hint in a SQL statement, the SQL statement is routed to the master instance if the read weight of the master instance is not 0.

<table>
<thead>
<tr>
<th>Specification code</th>
<th>Specification type</th>
<th>Memory</th>
<th>CPU</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysqlro.x8.xlarge.1</td>
<td>Dedicated instance</td>
<td>64 GB</td>
<td>8</td>
<td>800</td>
</tr>
<tr>
<td>mysqlro.x4.2xlarge.1</td>
<td>Dedicated instance</td>
<td>64 GB</td>
<td>16</td>
<td>800</td>
</tr>
<tr>
<td>mysqlro.x8.2xlarge.1</td>
<td>Dedicated instance</td>
<td>128 GB</td>
<td>16</td>
<td>1600</td>
</tr>
<tr>
<td>mysqlro.x4.4xlarge.1</td>
<td>Dedicated instance</td>
<td>128 GB</td>
<td>32</td>
<td>1600</td>
</tr>
<tr>
<td>rds.mysql.st.d13</td>
<td>Dedicated-host instance</td>
<td>220 GB</td>
<td>30</td>
<td>3000</td>
</tr>
<tr>
<td>rds.mysql.st.h43</td>
<td>Dedicated-host instance</td>
<td>470 GB</td>
<td>60</td>
<td>6000</td>
</tr>
</tbody>
</table>
For example, after a hint is prefixed to the following statement, the statement is always routed to and executed on the master instance regardless of the preset weight:

```sql
/*FORCE_MASTER*/ SELECT * FROM table_name;
```

### 19.5.8 Verify the allocated read weights

This topic describes how to verify the allocated read weights when read/write splitting is enabled.

To verify the load ratio of each read weight, you can run the `select @@server_id;` command for 10,000 times by using persistent connections and collect the number of times that each `server_id` appears in the output.

Alternatively, you can check whether the load ratio of read weight is consistent with the distributed ratio by using one of the following two methods:

**Verify the load ratio based on monitoring data in the console**

1. Log on to the [RDS console](https://console.aliyun.com/).
2. In the upper-left corner, select the region where the target RDS instance is located.
3. Find the target RDS instance and click the instance ID.
4. In the left-side navigation pane, click **Monitoring and Alerts**.
5. On the **Monitoring** tab, select the monitoring type **Engine Monitoring**.
6. View the **TPS (Transactions per Second)/QPS (Queries per Second)** chart to obtain the number of read and write operations on each of the master and read-only instances.

**Note:**

Refreshing the TPS/QPS chart takes about five minutes.

7. Compare the QPS/TPS of each instance to verify whether the load ratio is correct.
Verify the SQL load by directly connecting to each instance

You can view the number of SQL statements executed by each instance by connecting to the master database and read-only databases involving read/write splitting.

Note:
To verify this, the connection addresses of the master database and read-only databases instead of the read/write splitting address are needed.

Run either of the following commands to verify the SQL load:

```sql
select * from information_schema.global_status where VARIABLE_NAME = 'COM_SELECT';
```

```sql
select * from information_schema.global_status where VARIABLE_NAME = 'COM_INSERT';
```

19.5.9 Verify the effect of read/write splitting

19.5.9.1 Use SQL Explorer to verify the effect of read/Write splitting

This topic describes how to use SQL Explorer to verify the effect of read/write splitting by comparing the number of SQL statements executed by each of the master and read-only instances.

For information about how to enable SQL Explorer, see SQL Explorer.

19.5.9.2 Use internal SQL statements to verify the effect of read/write splitting

This topic describes how to verify the effect of read/write splitting for an RDS MySQL instance by running the `/*PROXY_INTERNAL*/show last route;` command.

Note:
Currently, do not use this SQL statement in the production environment because it is still being tested internally and may be adjusted later as required.

View the database to which an SQL statement is sent for execution

You can view the ID of the instance on which an SQL statement is executed by running the following command:

`/*PROXY_INTERNAL*/show last route;`

Note:
ApsaraDB for RDS

RDS provides a built-in hint SQL statement (which can only be executed by using read/write splitting VIP). When running this statement on the MySQL client, the -c option must be selected. Otherwise, the client filters out the hit and the following error is returned:

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'last route' at line 1
```

The returned result last_bkid indicates the ID of the database to which the last SQL statement (the one before the hit) was sent. This ID is the unique identification of an RDS instance. The following figure shows the details.

![MySQL command output](image)

**Note:**

Given that the SQL load for RDS is measured in batches with a minimum unit of 100 entries, the first 100 select statements are executed on the same instance before the 101st select statement is routed to another instance. To verify this, you can write a simple SQL file such as a.sql:

```
select 1;
/*PROXY_INTERNAL*/show last route;select 1;
```
Now you can see that the 101st SQL statement is routed to another instance (assuming that more than two read-only instances are available for sharing the load).

**Verify that all write requests are forwarded to the master database (master instance) for execution**

Once read/write splitting is enabled for RDS instances, write requests can only be forwarded to the master database because read-only databases only process read requests. Even when a system or routing error occurs, for example, a write SQL statement is routed to a read-only database, such write request can be routed back to the master database for execution according to the error reason (read_only error).

Additionally, you can run the insert statement, and then run the following hint SQL statement to verify that all write requests are forwarded to the master database.

```sql
/*PROXY_INTERNAL*/show last route;
```

**19.5.10 FAQ for read/write splitting**

This topic describes the FAQ related to read/write splitting.

**Why does the new read weights not take effect?**

After you change the read weights, RDS allocates read and write requests based on the new weights only after a new connection is established. However, your application does not terminate the existing connection or reestablish a connection.

**Why does the load on each of the master and read-only instances not comply with the specified read weight?**

If the load on each of the master and read-only instances does not comply with the specified read weight, check the following two aspects:

- Whether the request statements contain transactions. All requests that contain transactions, including transaction read requests, are routed only to the master instance.
- Whether your application connects to each of the master and read-only instances only by using the read/write splitting address. If your application connects to any master or read-only instance by using the endpoint of this specific instance, RDS does not allocate requests to this instance based on the specified weight.
Why does RDS route read requests to the master instance when the read weight of the master master is 0?

Check the following two aspects:

• Whether the request statements contain transactions. All requests that contain transactions, including transaction read requests, are routed only to the master instance.

• Whether the unavailability duration or replication latency of each read-only instance with a specified read weight exceeds the preset threshold. If so, RDS determines that all the read-only instances are unavailable and therefore routes read requests to the master instance.

How does read/write splitting guarantee the timeliness of data reading?

For more information, see How does read/write splitting ensure the timeliness of data reading?.

How do I set read weights by calling API actions?

For more information, see #unique_329.

How do I select the network type of the read/write splitting address?

If your application is deployed in the same VPC as the RDS instance, we recommend that you use the internal endpoint to guarantee data security and communication efficiency. If your application is deployed on your computer or in a different network from the RDS instance, you can use the public endpoint to establish a connection. For more information about how to switch between network types, see Change the network type of the read/write splitting address for an RDS MySQL instance.

What is the difference between the read/write splitting address and the internal and public endpoints of the master instance?

After read/write splitting is enabled, a read/write splitting address is generated. You must add the read/write splitting address and the read weight of each instance to the configuration data in your application. Then, after the requests from the application reach this address, RDS automatically allocates them to each of the master and read-only instances based on the request types and the read weights.

If the connection address in your application is the internal or public endpoint of the master instance, then all the requests are routed to the master instance.
Can I change the read/write splitting address?

No. You cannot change the read/write splitting address. The read/write splitting address is fixed. It does not change as you enable or disable read/write splitting, therefore you do not need to modify the configuration data in your application frequently. This reduces maintenance costs.

Does adding a read-only instance have any impact on the master instance?

No. Adding a read-only instance does not have any impact on the master instance because the read-only instance replicates data from the slave instance.
20 Account

20.1 Create an account for an ApsaraDB RDS for MySQL instance

This topic describes how to create an account that is used to manage databases on an ApsaraDB RDS for MySQL instance.

You can create RAM users under your Alibaba Cloud account and grant them permissions on specific RDS instances. For more information, see Create a RAM user.

Prerequisites

An ApsaraDB RDS for MySQL instance is created. For more information, see Create an ApsaraDB RDS for MySQL instance.

Account types

ApsaraDB RDS for MySQL supports two types of database accounts: privileged and standard. You can manage all your accounts and databases in the ApsaraDB for RDS console. For more information about the permissions that can be granted to each type of account, see Account permissions.

Note:
The type of an account cannot be changed. You can delete an account and then create a new one with the same account name. For more information, see Delete an account for an RDS MySQL instance.
<table>
<thead>
<tr>
<th>Account type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Privileged account | • You can create and manage privileged accounts by using the ApsaraDB for RDS console or API operations.  
• You can create only one privileged account per instance and then use the privileged account to manage all standard accounts and databases on that instance.  
• A privileged account enables you to manage permissions at a finer level. For example, you can grant query permissions on specific tables to standard accounts.  
• A privileged account has all permissions on all the databases of the instance on which it is created.  
• A privileged account has permissions to disconnect all standard accounts on the instance on which it is created. |
| Standard account    | • You can create and manage standard accounts by using the ApsaraDB for RDS console, API operations, or SQL statements.  
• You can create more than one standard account per instance. The maximum number of standard accounts allowed varies based on the database engine kernel you use.  
• You must manually grant permissions on specific databases to standard accounts.  
• A standard account does not have permissions to create, manage, or disconnect other accounts on the instance on which it is created. |

<table>
<thead>
<tr>
<th>Account type</th>
<th>Number of databases</th>
<th>Number of tables</th>
<th>Number of accounts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privileged account</td>
<td>Unlimited</td>
<td>&lt; 200,000</td>
<td>Varies based on the kernel parameter settings of the instance</td>
</tr>
<tr>
<td>Standard account</td>
<td>500</td>
<td>&lt; 200,000</td>
<td>Varies based on the kernel parameter settings of the instance</td>
</tr>
</tbody>
</table>

Create a privileged account

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID.

4. In the left-side navigation pane, click Accounts.

5. Click Create Account.

6. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Database Account   | Enter the username of the account. The username must meet the following requirements:  
  • The username must be 2 to 16 characters in length.  
  • The username must start with a letter and end with a letter or digit.  
  • The username can contain lowercase letters, digits, and underscores (_).  
  • The username cannot be the same as the username of an existing account. |
| Account Type       | Select Privileged Account. |
| Password           | Enter the password of the account. The password must meet the following requirements:  
  • The password must be 8 to 32 characters in length.  
  • The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.  
  • Special characters include ! @ # $ % ^ & * ( ) _ + - = |
| Confirm Password   | Enter the password of the account again. |
| Description        | Enter a description that helps identify the account. The description can be up to 256 characters in length. |
7. Click **OK**.

**Reset the permissions of a privileged account**

If the privileged account of your RDS instance encounters an exception, such as its permissions are accidentally revoked, follow these steps to restore the permissions:

1. Log on to the **ApsaraDB for RDS console**.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click **Accounts**.
5. Find the privileged account, and click **Reset Permissions** in the **Actions** column.
6. Enter the password of the privileged account to reset the account permissions.

**Create a standard account**

1. Log on to the **ApsaraDB for RDS console**.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click **Accounts**.
5. Click **Create Account**.
6. Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database Account</strong></td>
<td>Enter the username of the account. The username must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• The username must be 2 to 16 characters in length.</td>
</tr>
<tr>
<td></td>
<td>• The username must start with a letter and end with a letter or digit.</td>
</tr>
<tr>
<td></td>
<td>• The username can contain lowercase letters, digits, and underscores (_).</td>
</tr>
<tr>
<td><strong>Account Type</strong></td>
<td>Select <strong>Standard Account</strong>.</td>
</tr>
<tr>
<td><strong>Authorized Databases</strong></td>
<td>Select one or more databases on which you want to grant permissions to the account. You can leave this parameter empty, because you have the option to grant the account the permissions on specific databases after you create the account.</td>
</tr>
<tr>
<td></td>
<td>a. Select one or more databases from the Unauthorized Databases box and click <strong>Add</strong> to add them to the Authorized Databases box.</td>
</tr>
<tr>
<td></td>
<td>b. In the Authorized Databases box, select the <strong>Read/Write</strong>, <strong>Read-only</strong>, <strong>DDL Only</strong>, or <strong>DML Only</strong> permissions on each authorized database.</td>
</tr>
<tr>
<td></td>
<td>If you want to grant the same permissions on more than one authorized database simultaneously, select the authorized databases and click the button in the upper-right corner. For example, click <strong>Set All to Read/Write</strong>.</td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>Enter the password of the account. The password must meet the following requirements:</td>
</tr>
<tr>
<td></td>
<td>• The password must be 8 to 32 characters in length.</td>
</tr>
<tr>
<td></td>
<td>• The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.</td>
</tr>
<tr>
<td></td>
<td>• Special characters include ! @ # $ % ^ &amp; * ( ) _ + - =</td>
</tr>
<tr>
<td><strong>Confirm Password</strong></td>
<td>Enter the password of the account again.</td>
</tr>
</tbody>
</table>
### ApsaraDB for RDS

#### RDS MySQL Database / 20 Account

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Optional. Enter a description that helps identify the account. The description can be up to 256 characters in length.</td>
</tr>
</tbody>
</table>

7. Click **OK**.

### FAQ

- **Can I configure an account so it is only accessible from an internal network?**
  
  This configuration is not supported in the ApsaraDB for RDS console. You must use SQL statements to specify the IP addresses from which an account can log on. For more information, see Limit permissions of a specific IP address on a database.

- **Can I manage accounts at finer levels such as the table level?**
  
  This configuration is not supported in the ApsaraDB for RDS console. You must use SQL statements to manage accounts at finer levels. For more information, see Authorize accounts to manage tables, views, and fields.

### Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_98</td>
<td>Creates an account to manage databases on an ApsaraDB for RDS instance.</td>
</tr>
</tbody>
</table>

### 20.2 Reset the password of an account for an RDS MySQL instance

This topic describes how to reset the password of an account for an RDS MySQL instance in case that the password is lost.

**Note:**

For data security purposes, we recommend you change the password on a regular basis.

**Procedure**

1. Log on to the RDS console.
2. In the upper-left corner, select the region where the target RDS instance is located.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click Accounts.

5. On the Accounts tab, select the account whose password you want to reset, and in the Actions column click Reset Password.

6. In the Reset Account Password dialog box, enter a new password and confirm it, then click OK.

**Note:**
The password must meet the following requirements:

- The password must be 8 to 32 characters in length.
- The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters.
- Special characters include ! @ # $ % ^ & * ( ) _ + - =

**APIs**

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_335</td>
<td>Used to reset the password of a database account.</td>
</tr>
</tbody>
</table>
20.3 Modify the permissions of an account for an RDS instance

You can edit the permissions of a standard account as needed. The permissions of privileged accounts can only be reset to the default settings and cannot be changed to a specific set of permissions.

Modify the permissions of a standard account

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner, select the region where the target RDS instance is located.
3. Find the target RDS instance and click the instance ID.
4. In the left-side navigation pane, click Accounts.
5. Find the account for which you want to modify permissions, click Edit permissions.
6. In Modify Account Permissions page and adjust account permissions.
   - Modify the database to authorize: Select the database and click > or <.
   - Set read and write permissions in Authorized Databases, you can set permissions to Read/Write(DDL+DML), Read-only, DDL only or DML only.

Note:
If you need more granular permissions, use SQL commands to make permission changes. For more information about permissions, see Account permissions.

Modify Account Permissions

Database Account: [Account Name]

Authorized Databases:

Unauthorized Databases

[rds.label.mysql AuthorizedDatabase]

- Read/Write (DDL + DML)
- Read-only

2 Items

7. Click **OK**.

20.4 Reset the permissions of a premier account

This topic describes how to reset the permissions of a premier account when the premier account becomes abnormal, for example, when the permissions of the premier account are revoked unexpectedly.

Procedure

1. Log on to the RDS console.

2. In the upper-left corner, select the region where the target RDS instance is located.
3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click Accounts.

5. On the Accounts tab, find the Privileged Account and in the Actions column click Reset Permissions.

6. Enter the password of the premier account and click OK.

20.5 Authorize a service account for an RDS MySQL instance

This topic describes how to authorize a service account for an RDS MySQL instance.

If you are seeking for technical support from Alibaba Cloud and if it is necessary to operate your DB instance during technical support, you must authorize a service account that is used by the technical support staff to provide technical support services.

Prerequisites

This function is available only to the following DB engine versions and editions:

- MySQL 8.0 High-availability Edition (with local SSDs)
- MySQL 5.7 Enterprise Edition
- MySQL 5.7 High-availability Edition (with local SSDs)
- MySQL 5.6 High-availability Edition
- MySQL 5.5 High-availability Edition

Procedure

1. Log on to the RDS console.
2. In the left-side navigation pane, select the region where the target RDS instance is located.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click Accounts.

5. On the Service Account Permissions tab, select the permission to be authorized to the service account and in the Privilege Status column click the switch.

   - For troubleshooting of the IP address whitelists, database parameters, and other problems, you only need to authorize the Configuration Permission.
   - For the database performance problems caused by your application, you must authorize the Data Permission.

If you seek technical support, you must allow our support engineers to use your service account to log on and perform operations.
6. In the **Set Expiration Time** dialog box, set the permission expiration time and click **OK**.

![Set Expiration Time dialog box](image)

**What to do next**

After you authorize permissions to a service account, you can cancel the authorization or change the authorization validity period on the **Service Account Permissions** tab.

![Service Account Permissions tab](image)

---

20.6 Delete an account for an RDS MySQL instance

This topic describes how to delete an account in the RDS console or run an SQL statement to delete a standard account.

**Use the console to delete an account**

1. Log on to the **RDS console**.
2. In the upper-left corner, select the region where the target RDS instance is located.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click Accounts.

5. On the Accounts tab, find the account you want to delete, and in the Actions column click Delete.

6. In the displayed dialog box, click Confirm.

Run an SQL statement to delete a standard account

You can only run an SQL statement to delete a standard account for an RDS instance.

1. Log on to the target RDS instance. For more information, see How to connect to ApsaraDB?

2. Run the following command to delete a standard account:

```
DROP USER 'username'@'localhost';
```

APIs

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_336</td>
<td>Used to delete an account for an RDS instance.</td>
</tr>
</tbody>
</table>

20.7 Limit permissions of a specific IP address on a database

This topic describes how to limit the permissions of a specific IP address on a database of an ApsaraDB RDS for MySQL instance. If a whitelist of an RDS instance contains more than one IP address, a database account created in this instance has the same permissions after logging on to this instance from one of these IP addresses. This imposes security risks. For example, if a whitelist contains the IP addresses of both the headquarters and branches of an enterprise, you can grant a database account only the permissions to access and manage a database from the IP address of the headquarters.
Prerequisites

A privileged account is created in your ApsaraDB RDS for MySQL instance. For more information, see Create an account for an RDS MySQL instance.

Procedure

1. Log on to the ApsaraDB for RDS console.

2. In the upper-left corner of the page, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID.

4. In the upper-right corner, click Log On to DB. Then, use the privileged account to log on to the target database in the RDS instance. For more information, see Log on to an RDS MySQL instance through DMS.

5. In the top navigation bar, choose SQL Operations > SQL Window.

6. Create a user and grant that user permissions to access and manage a database from a specific IP address. Note that you cannot view the Authorized Databases of the created user in the ApsaraDB for RDS console.

Example:

Execute the following statements to create a user named test001 and grant that user permissions to access and manage the rds001 database from the IP address 42.120.74.119:

```
CREATE USER `test001`@`42.120.74.119` IDENTIFIED BY `passwd`;
GRANT PROCESS, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO `test001`@`42.120.74.119`;
GRANT ALL PRIVILEGES ON `rds001`.* TO `test001`@`42.120.74.119`;
GRANT SELECT ON `mysql`.`help_topic` TO `test001`@`42.120.74.119`;
GRANT SELECT ON `mysql`.`time_zone` TO `test001`@`42.120.74.119`;
GRANT SELECT ON `mysql`.`slow_log` TO `test001`@`42.120.74.119`;
GRANT SELECT ON `mysql`.`time_zone_transition` TO `test001`@`42.120.74.119`;
GRANT SELECT ON `mysql`.`event` TO `test001`@`42.120.74.119`;
GRANT SELECT ON `mysql`.`proc` TO `test001`@`42.120.74.119`;
GRANT SELECT ON `mysql`.`help_category` TO `test001`@`42.120.74.119`;
GRANT SELECT ON `mysql`.`help_relation` TO `test001`@`42.120.74.119`;
GRANT SELECT ON `mysql`.`help_keyword` TO `test001`@`42.120.74.119`;
GRANT SELECT ON `mysql`.`general_log` TO `test001`@`42.120.74.119`;
GRANT SELECT ON `mysql`.`time_zone_leap_second` TO `test001`@`42.120.74.119`;
GRANT SELECT ON `mysql`.`time_zone_transition_type` TO `test001`@`42.120.74.119`;
GRANT SELECT ON `mysql`.`time_zone_name` TO `test001`@`42.120.74.119`;
```

Note:
• If you change the IP address 42.120.74.119 in all the preceding statements to the wildcard %, the created user functions the same as a database account you create in the ApsaraDB for RDS console. This means that you can view the Authorized Databases of the created user in the ApsaraDB for RDS console.

• To change the IP address to 42.120.74.120, you can execute the following statement:

```
RENAME USER `test001`@`42.120.74.119` TO `test001`@`42.120.74.120`;
```

### 20.8 Authorize accounts to manage tables, views, and fields

This topic describes how to execute SQL statements for authorizing accounts to manage tables, views, or fields in a database of an ApsaraDB for RDS instance. If you are using your Alibaba Cloud account, you have the permissions to manage all types of data in your ApsaraDB for RDS instance.

**Prerequisites**

You have created a privileged account in your ApsaraDB for RDS instance. For more information, see [Create accounts and databases for an RDS for MySQL instance](#).

**Procedure**

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where your RDS instance resides.
3. Find your RDS instance and click its ID.
4. In the upper-right corner, click **Log On to DB**. Then, use this privileged account to log on to the target database in your RDS instance.
5. From the top navigation bar, choose **SQL Operations > SQL Window**.
6. Execute SQL statements to create an account and authorize it to manage tables, views, and fields in the target database.

**Note:**

The created account does not have permissions to view its authorized databases in the ApsaraDB for RDS console.

- Create an account and authorize it to manage a table in the target database.

```
CREATE USER `<The username of the account you want to create>`@`%` IDENTIFIED BY `'<The password of the account you want to create>'; GRANT PROCESS, REPLICATION SLAVE, REPLICATION CLIENT ON `*.*` TO `'<The username of the created account'>`@`'%';
```
Example:

To create an account named test01 and authorize it to manage the test100 table in the rds001 database, execute the following SQL statements:

```
CREATE USER `test01`@`%` IDENTIFIED BY `passwd`;
GRANT PROCESS, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO `test01`@`%`;
GRANT ALL PRIVILEGES ON `rds001`.`test100` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`help_topic` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`func` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`time_zone` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`slow_log` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`time_zone_transition` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`event` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`proc` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`help_category` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`help_relation` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`help_keyword` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`general_log` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`time_zone_leap_second` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`time_zone_transition_type` TO `test01`@`%`;
GRANT SELECT ON `mysql`.`time_zone_name` TO `test01`@`%`;
```
If you change test100 in the third line of code to the * wildcard, you will give the test01 user the permissions to view its Authorized Databases in the ApsaraDB for RDS console.

- Authorize the created account to query a view from the target database.

```sql
grant select on <The name of the target database>.<The name of the view you want to query with the created account> to <The username of the created account>;
```

Example:

To authorize the test01 user to query the view_test1 view from the rds001 database, execute the following SQL statement:

```sql
grant select on rds001.view_test1 to test01;
```

- Authorize the created account to update or query a field name in a table from the target database.

```sql
grant update (<The field name you want to update with the created account>) on table <The name of the table where the field name you want to update resides> to <The username of the created account>;
```

```sql
grant select (<The field name you want to query with the created account>) on table <The name of the table where the field name you want to query resides> to <The username of the created account>;
```

Example:

To authorize the test01 user to update the testid field in the testtable table, execute the following SQL statement:

```sql
grant update (testid) on table testtable to test01;
```

In addition to executing SQL statements, you can choose Tools > User Management from the top navigation bar to change the permissions of an account.

### 20.9 System accounts

ApsaraDB RDS for MySQL provides various types of system accounts. However, you do not need to be concerned about the permissions and operations of these system accounts. This topic describes these system accounts.

<table>
<thead>
<tr>
<th>Account name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>root (aliyun_root in MySQL 5.7 and later)</td>
<td>The local O&amp;M account that you can use to manage instances, such as when you modify kernel parameters and query instance status.</td>
</tr>
<tr>
<td>Account name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>aurora and rds_service</td>
<td>The accounts that you can use to remotely access and manage instances, such as when you perform a primary/secondary switchover and monitor instances.</td>
</tr>
<tr>
<td>aurora_proxy</td>
<td>The account that you can use to forward connections after the database proxy is enabled.</td>
</tr>
</tbody>
</table>

### 20.10 Account permissions

This topic provides an overview of the permissions that ApsaraDB RDS for MySQL provides for both privileged and standard accounts.

#### Accounts and permissions

<table>
<thead>
<tr>
<th>Account type</th>
<th>Permission</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privileged accounts</td>
<td>SELECT</td>
<td>INSERT</td>
</tr>
<tr>
<td></td>
<td>DROP</td>
<td>RELOAD</td>
</tr>
<tr>
<td></td>
<td>ALTER</td>
<td>CREATE TEMPORARY TABLES</td>
</tr>
<tr>
<td></td>
<td>REPLICATION CLIENT</td>
<td>CREATE VIEW</td>
</tr>
<tr>
<td></td>
<td>CREATE USER</td>
<td>EVENT</td>
</tr>
<tr>
<td>Standard accounts</td>
<td>SELECT</td>
<td>LOCK TABLES</td>
</tr>
<tr>
<td></td>
<td>REPLICATION CLIENT</td>
<td></td>
</tr>
<tr>
<td>Read/write account</td>
<td>SELECT</td>
<td>INSERT</td>
</tr>
<tr>
<td></td>
<td>DROP</td>
<td>REFERENCES</td>
</tr>
<tr>
<td></td>
<td>LOCK TABLES</td>
<td>EXECUTE</td>
</tr>
<tr>
<td></td>
<td>ALTER ROUTINE</td>
<td>EVENT</td>
</tr>
<tr>
<td>Account type</td>
<td>Permission</td>
<td>Operation</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>-----------</td>
</tr>
<tr>
<td>DDL-only</td>
<td>CREATE</td>
<td>DROP</td>
</tr>
<tr>
<td></td>
<td>INDEX</td>
<td>ALTER</td>
</tr>
<tr>
<td></td>
<td>CREATE</td>
<td>TEMPORARY TABLES</td>
</tr>
<tr>
<td></td>
<td>LOCK TABLES</td>
<td>CREATE VIEW</td>
</tr>
<tr>
<td></td>
<td>SHOW VIEW</td>
<td>CREATE ROUTINE</td>
</tr>
<tr>
<td></td>
<td>ALTER</td>
<td>ROUTINE</td>
</tr>
<tr>
<td></td>
<td>PROCESS</td>
<td>REPLICATION</td>
</tr>
<tr>
<td></td>
<td>SLAVE</td>
<td>CLIENT</td>
</tr>
<tr>
<td>DML-only</td>
<td>SELECT</td>
<td>INSERT</td>
</tr>
<tr>
<td></td>
<td>UPDATE</td>
<td>DELETE</td>
</tr>
<tr>
<td></td>
<td>CREATE</td>
<td>TEMPORARY TABLES</td>
</tr>
<tr>
<td></td>
<td>LOCK TABLES</td>
<td>EXECUTE</td>
</tr>
<tr>
<td></td>
<td>SHOW VIEW</td>
<td>EVENT</td>
</tr>
<tr>
<td></td>
<td>TRIGGER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PROCESS</td>
<td>REPLICATION</td>
</tr>
<tr>
<td></td>
<td>SLAVE</td>
<td>CLIENT</td>
</tr>
</tbody>
</table>
21 Database

21.1 Create a database for an RDS MySQL instance

This topic describes how to create a database for an RDS MySQL instance.

Procedure

1. Log on to the RDS console.
2. In the upper-left corner, select the region where the target RDS instance is located.
3. Find the target RDS instance and click the instance ID.
4. In the left-side navigation pane, click Databases.
5. Click Create Database.
6. Set the following parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>The account name contains 2 to 16 characters, including lowercase letters, digits, underscores (_), and hyphens (-). It must begin with a letter and end with a letter or digit. Each database name must be unique in the RDS instance.</td>
</tr>
<tr>
<td>Supported Character Set</td>
<td>Select utf8, gbk, latin1, or utf8mb4. If you need another character set, select all and then select the character set from the list.</td>
</tr>
</tbody>
</table>
**Authorized Account**
Select the account that needs to access this database. You can also leave this parameter blank and set the authorized account after the database is created.

**Note:**
Only standard accounts are displayed, because the premier account has all permissions for all databases.

**Account Type**
Select **Read/Write**, **Read-only**, **DDL only**, or **DML only**.

**Description**
Optional. Enter the other account information that helps to better manage the account. You can enter up to 256 characters.

7. Click **OK**.

### APIs

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_339</td>
<td>Used to create a database for an RDS instance.</td>
</tr>
</tbody>
</table>

### 21.2 Delete a database for an RDS MySQL instance

This topic describes how to delete a database for an RDS MySQL instance by using the RDS console or running an SQL statement. Each method applies to different types of instances. You can choose a suitable method based on the RDS instance whose database you want to delete.

**Delete a database by using the RDS console**

1. Log on to the **RDS console**.

2. In the upper-left corner, select the region where the target RDS instance is located.

3. Find the target RDS instance and click the instance ID.
4. In the left-side navigation pane, click **Databases**.

5. Find the database you want to delete, and in the **Actions** column click **Delete**.

6. In the displayed dialog box, click **Confirm**.

**Delete a database by running an SQL statement**

1. Log on to the **RDS console**.

2. In the upper-left corner, select the region where the target RDS instance is located.

3. Find the target RDS instance and click the instance ID.

4. In the upper-right corner, click **Log On to DB** go to the **RDS Database Logon** page of the **DMS console**.

5. On the **RDS Database Logon** page, check the connection address and port information.

If the information is correct, enter the database username and password.

Parameter description:

- **Databases Username**: the name of the premier account.
- **Password**: the password for the premier account.

**Note:**
You can view the connection address and port information of this account on the **Basic Information** page of the instance in the RDS console.

6. Enter the verification code and click **Log On**.

**Note:**
If you want the browser to remember the password, select **Remember Password** before you click **Log On**.

7. If DMS prompts you to add the IP address segment of the DMS server to an IP address whitelist of the RDS instance, click **Configure Whitelist**. For more information about how to manually configure the whitelist, see **Configure a whitelist for an RDS MySQL instance**.

Issue: 20200702
8. Click Log On.

9. In the top navigation bar, choose SQL Operations > SQL Window.

10. Run the following command to delete a database:

    DROP DATABASE <database name>;

11. Click execute to delete the database.

**APIs**

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_341</td>
<td>Used to delete a database for an RDS instance.</td>
</tr>
</tbody>
</table>
22 Data security

22.1 Switch to the enhanced whitelist mode for an RDS MySQL instance

This topic describes how to switch from the standard whitelist mode to the enhanced whitelist mode for an RDS MySQL instance.

IP whitelist modes

ApsaraDB for RDS instances provide the following two IP whitelist modes:

- Standard whitelist mode

In this mode, the IP addresses in the whitelist do not distinguish between classic networks and VPCs. The IP addresses in the whitelist can access the RDS instance both in classics network and VPCs. We recommend that you switch from the standard whitelist to the enhanced whitelist.

- Enhanced whitelist mode

In this mode, the whitelist is classified into two IP whitelist groups by network type: the classic-network whitelist group and the VPC whitelist group. When you create an IP whitelist, you must specify a network type.
Changes after switching to the enhanced whitelist mode

- If the network type of the instance is VPC, a new whitelist of the VPC is generated and contains the same IP addresses in the original whitelist. The new IP whitelist group only applies to VPCs.
- If the instance network type is classic network, a new whitelist group is generated and contains the same IP addresses in the original whitelist. The new IP whitelist group only applies to classic networks.
- If the instance is in the hybrid access mode, two new whitelist groups are generated and each contains the same IP addresses in the original whitelist. One of the whitelist group applies to VPCs and the other applies to classic networks.

**Note:**
Switching to the enhanced whitelist mode does not affect the ECS instances that are in the ECS security group whitelist.

Prerequisites

The DB version used by the instance is one of the following:

- MySQL 5.7 High-Availability Edition
- MySQL 5.6
- MySQL 5.5

Precautions

- You can switch from the standard whitelist mode to the enhanced whitelist mode. However, you cannot switch from the enhanced whitelist mode to the standard whitelist mode.
- In the enhanced whitelist mode, the classic-network whitelist group also applies to accesses from a public network. If you want to access the RDS instance from an instance, host, or application in the public network, you must add the public IP address to the classic-network whitelist group.

Procedure

1. Log on to the RDS console.
2. In the upper-left corner, select the region where the target RDS instance is located.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click Data Security.

5. On the Whitelist Settings tab, click Switch to Enhanced Whitelist (Recommended).

6. In the message box that appears, click OK.

22.2 Configure a whitelist for an RDS MySQL instance

This topic describes how to configure a whitelist for an RDS MySQL instance.

For more information about how to configure a whitelist in other database engines, see the following topics:

- Configure a whitelist for an RDS SQL Server instance
- Configure a whitelist for an RDS PostgreSQL instance
- Configure a whitelist for an RDS PPAS instance
- Configure a whitelist for an RDS MariaDB instance

Background information

ApsaraDB RDS MySQL provides two types of whitelists:
• IP address whitelists

An IP address whitelist contains the IP addresses of entities that require access to your RDS instance. The IP address whitelist labeled default contains only the default IP address 127.0.0.1, which indicates that all entities are denied access to your RDS instance.

Before you configure an IP address whitelist, you must confirm the network isolation mode of your RDS instance. The configuration procedure varies depending on the network isolation mode used.

- **Standard whitelist mode**

  In standard whitelist mode, an IP address whitelist can contain IP addresses from both classic networks and VPCs. The standard whitelist mode may incur security risks. We recommend that you switch the network isolation mode from standard whitelist to enhanced whitelist.

- **Enhanced whitelist mode**

  In enhanced whitelist mode, an IP address whitelist can contain only IP addresses from classic networks or VPCs. When you create an IP address whitelist, you must specify its network type.
Security groups

A security group serves as a virtual firewall to limit the inbound and outbound traffic of ECS instances in that security group. After you add a security group, all ECS instances in it are granted access to your RDS instance.

For more information, see Create a security group.

Whitelists make your RDS instance more secure, and the configuration process does not interrupt the operation of your RDS instance. Therefore, we recommend that you maintain whitelists on a regular basis.

Precautions for configuring an IP address whitelist

- You can edit or clear a default IP address whitelist, but cannot delete it.
- Each IP address whitelist can contain up to 1,000 IP addresses or CIDR blocks. If you want to add more than 1,000 IP addresses, we recommend that you combine them into CIDR blocks such as 192.168.1.0/24.
- If you attempt to log on to Data Management Service (DMS) from your RDS instance without adding the IP address of DMS to a whitelist, DMS displays a message that states your logon can succeed only after its IP address is added to a whitelist. By default, DMS automatically creates an IP address whitelist that contains its IP address in your RDS instance.

Configure an IP address whitelist in enhanced whitelist mode

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click Data Security.
5. Confirm the connection scenario and perform its required operations.

<table>
<thead>
<tr>
<th>Connection scenario</th>
<th>Operation</th>
</tr>
</thead>
</table>
| (Recommended)  Your ECS and RDS instances reside in the same VPC. | a. On the **Whitelist Settings** tab of the Data Security page, click **Edit** to the right of the IP address whitelist labeled **default VPC**.  
   b. In the dialog box that appears, enter the internal IP address of your ECS instance in the Whitelist field and click **OK**.  
   ![Note:](image) Applications running on your ECS instance connect to the internal endpoint of your RDS instance. |
| Your ECS and RDS instances reside in different VPCs.     | a. Navigate to the **Database Connection** page and click **Switch to Classic Network**. In the dialog box that appears, click **OK**.  
   b. Click **Switch to VPC**. In the dialog box that appears, select the VPC that houses your ECS instance and click **OK**.  
   ![Note:](image) Your ECS and RDS instances can be switched to the same VPC only if they reside in the same region. If they reside in different regions, we recommend that you use Data Transmission Service (DTS) to migrate your RDS instance to the region where your ECS instance resides. This helps ensure service availability. For more information, see [Migrate between RDS instances](#).  
   c. Navigate to the **Whitelist Settings** tab of the Data Security page, and click **Edit** to the right of the IP address whitelist labeled **default VPC**.  
   d. In the dialog box that appears, enter the internal IP address of your RDS instance in the Whitelist field and click **OK**.  
   ![Note:](image) Applications running on your ECS instance connect to the internal endpoint of your RDS instance. |
<table>
<thead>
<tr>
<th>Connection scenario</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your ECS and RDS instances both reside in classic</td>
<td><strong>a.</strong> Navigate to the <a href="#">Whitelist Settings</a> tab of the Data Security page, and click <strong>Edit</strong> to the right of the IP address whitelist labeled <strong>default Classic Network</strong>.</td>
</tr>
<tr>
<td>networks.</td>
<td><strong>b.</strong> In the dialog box that appears, enter the internal IP address of your ECS instance in the Whitelist field and click <strong>OK</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Applications running on your ECS instance connect to the internal endpoint of your RDS instance.</td>
</tr>
<tr>
<td>Your ECS instance resides in a classic network.</td>
<td><strong>a.</strong> Migrate your ECS instance to the VPC that houses your RDS instance. For more information, see <a href="#">Migrate an ECS instance</a>.</td>
</tr>
<tr>
<td>Your RDS instance resides in a VPC.</td>
<td><strong>Note:</strong> Your ECS and RDS instances can be switched to the same VPC only if they reside in the same region. If they reside in different regions, we recommend that you use DTS to migrate your RDS instance to the region where your ECS instance resides. This helps ensure service availability. For more information, see <a href="#">Migrate between RDS instances</a>.</td>
</tr>
<tr>
<td></td>
<td><strong>b.</strong> Navigate to the <a href="#">Whitelist Settings</a> tab of the Data Security page, and click <strong>Edit</strong> to the right of the IP address whitelist labeled <strong>default VPC</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>c.</strong> In the dialog box that appears, enter the internal IP address of your ECS instance in the Whitelist field and click <strong>OK</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Applications running on your ECS instance connect to the internal endpoint of your RDS instance.</td>
</tr>
</tbody>
</table>
### Connection scenario

<table>
<thead>
<tr>
<th>Your ECS instance resides in a VPC. Your RDS instance resides in a classic network.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operation</strong></td>
</tr>
<tr>
<td>a. Navigate to the <strong>Database Connection</strong> page and click <strong>Switch to VPC</strong>. In the dialog box that appears, select the VPC that houses your ECS instance and click OK.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td>Your ECS and RDS instances can be switched to the same VPC only if they reside in the same region. If they reside in different regions, we recommend that you use DTS to migrate your RDS instance to the region where your ECS instance resides. This helps ensure service availability. For more information, see <a href="#">Migrate between RDS instances</a>.</td>
</tr>
<tr>
<td>b. Navigate to the <strong>Whitelist Settings</strong> tab of the Data Security page, and click <strong>Edit</strong> to the right of the IP address whitelist labeled <strong>default VPC</strong>.</td>
</tr>
<tr>
<td>c. In the dialog box that appears, enter the internal IP address of your ECS instance in the Whitelist field and click <strong>OK</strong>.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td>Applications running on your ECS instance connect to the internal endpoint of your RDS instance.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Your host that requires access to your RDS instance resides outside the cloud.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operation</strong></td>
</tr>
<tr>
<td>a. Navigate to the <strong>Whitelist Settings</strong> tab of the Data Security page, and click <strong>Edit</strong> to the right of the IP address whitelist labeled <strong>default Classic Network</strong>.</td>
</tr>
<tr>
<td>b. In the dialog box that appears, enter the public IP address of your host in the Whitelist field and click <strong>OK</strong>.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td>• Applications running on your host connect to the public endpoint of your RDS instance.</td>
</tr>
<tr>
<td>• For more information about how to obtain the public IP address of your host, see <a href="#">#unique_88</a>.</td>
</tr>
</tbody>
</table>

**Note:**

- On the Whitelist Settings tab of the Data Security page, you can click **Create Whitelist** and then in the Create Whitelist dialog box select **VPC** or **Classic Network/Public IP** for Network Type.
- If you enter the CIDR block 10.10.10.0/24 in the Whitelist field, all IP addresses in the 10.10.10.X format are granted access to your RDS instance.
If you enter more than one IP address or CIDR block in the Whitelist field, make sure that they are separated with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
- If you click **Add Internal IP Addresses of ECS Instances**, IP addresses of all ECS instances created in your Alibaba Cloud account are displayed.

**Add Internal IP Addresses of ECS Instances**

You can add 999 more entries.

Specified IP address: If you specify the IP address 192.168.0.1, this IP address is allowed to access the RDS instance.

Specified CIDR block: If you specify the CIDR block 192.168.0.0/24, the IP addresses ranging from 192.168.0.1 to 192.168.0.255 are allowed to access the RDS instance.

When you add multiple IP addresses or CIDR blocks, separate them by a comma (no space after the comma), for example, 192.168.0.1,192.168.0.0/24.

**How to Locate the Local IP Address**

**New whitelist entries take effect in 1 minute.**

**Edit Whitelist**

**Network Type:**

- [ ] VPC
- [ ] Classic Network/Public IP

You currently cannot configure network isolation settings. You must enable enhanced whitelists first before configuring network isolation settings.

**Whitelist Name**: default

**Whitelist**: 127.0.0.1
Configure an IP address whitelist in standard whitelist mode

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click Data Security.
5. On the Whitelist Settings tab, click Edit to the right of the IP address whitelist labeled default.

**Note:**
You can also click Create Whitelist to create an IP address whitelist.

6. In the Edit Whitelist dialog box, enter IP addresses or CIDR blocks in the Whitelist field and click OK.

- If you enter the CIDR block 10.10.10.0/24, all IP addresses in the 10.10.10.X format are granted access to your RDS instance.
- If you enter more than one IP address or CIDR block, make sure that they are separated with commas (,). Do not add spaces before or after the commas. Example: 192.168.0.1,172.16.213.9.
- If you click Add Internal IP Addresses of ECS Instances, IP addresses of all ECS instances created in your Alibaba Cloud account are displayed.

**Note:**
After you add IP addresses or CIDR blocks to the IP address whitelist labeled default, the system deletes the default IP address 127.0.0.1.

### Edit Whitelist

**Network Type:**
- VPC
- Classic Network/Public IP

You currently cannot configure network isolation settings. You must enable enhanced whitelists first before configuring network isolation settings.

**Whitelist Name**: default

**Whitelist**: 127.0.0.1

**Add Internal IP Addresses of ECS Instances**

You can add 999 more entries.

- Specified IP address: If you specify the IP address 192.168.0.1, this IP address is allowed to access the RDS instance.
- Specified CIDR block: If you specify the CIDR block 192.168.0.0/24, the IP addresses ranging from 192.168.0.1 to 192.168.0.255 are allowed to access the RDS instance.

### Cases

- Only the default IP address 127.0.0.1 is added to a whitelist in the Data Security > Whitelist Settings navigation path.

The default IP address 127.0.0.1 indicates that all entities are denied access. You must whitelist the IP addresses of entities that require access to your RDS instance.
• I added only one entry, 0.0.0.0, to an IP address whitelist, with the intention to grant all entities access to my RDS instance. Enter the CIDR block 0.0.0.0/0 instead.

**Note:**
This CIDR block indicates that all entities are granted access to your RDS instance. Exercise caution when adding this CIDR block.

• The system prompts me with IP address errors when my RDS instance is in enhanced whitelist mode.

For more information, see [Switch the IP whitelist mode from standard to enhanced](#).

- If your RDS instance resides in a VPC and is connected by using its internal endpoint, make sure that the internal IP address of your ECS instance is added to the IP address whitelist labeled **default VPC**.

- If your RDS instance resides in a classic network and is connected by using its internal endpoint, make sure that the internal IP address of your ECS instance is added to the IP address whitelist labeled **default Classic Network**.

- If your RDS instance resides in a VPC and is connected by using ClassicLink, make sure that the internal IP address of your ECS instance is added to the IP address whitelist labeled **default VPC**.

- If your RDS instance is connected over the Internet, make sure that the public IP address of your ECS instance is added to the IP address whitelist labeled **default Classic Network**. (The IP address whitelist labeled default VPC cannot be used to allow access from the Internet.)

• The public IP addresses you add to whitelists are not real egress IP addresses.

Possible reasons are as follows:

- Public IP addresses dynamically change.
- The tool or website you use to query public IP addresses yields inaccurate results.

For more information, see [#unique_88](#)

**Precautions for configuring a security group**

• You can configure a security group only if your RDS instance runs MySQL 5.6, MySQL 5.7, or MySQL 8.0.
• Your RDS instance can have both IP address whitelists and security groups at the same time. All IP addresses in the configured whitelists and all ECS instances in the configured security group can access your RDS instance.

• You can only add one security group.

**Configure a security group**

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID. The Basic Information page appears.
4. In the left-side navigation pane, click Data Security.

**Note:**
If a security group is followed by a VPC tag, the ECS instances in it reside in VPCs.

6. Select the security group you want to add. Then, click OK.

**FAQ**

• Does an IP address whitelist take effect immediately after it is configured?

An IP address whitelist takes effect approximately one minute after it is configured.

• Why do I find IP address whitelists that are not created by me?

If these whitelists contain internal IP addresses, they are probably generated by other Alibaba Cloud products such as DMS or DAS and do not invoke operations on your service data.

• Is my RDS instance exposed to security risks if I enable only internal network access and disable Internet access?

We recommend that you switch the network type of your RDS instance from Classic Network to VPC. After you make the switch, only ECS instances in the same VPC as your RDS instance are granted access on the condition that their internal IP addresses are added to whitelists. For more information, see Change the network type of an RDS MySQL instance.
22.3 Configure SSL encryption for an RDS MySQL instance

This topic describes how to enable Secure Sockets Layer (SSL) encryption and install SSL CA certificates to applications. SSL encrypts data over network connections at the transport layer. This enhances data security and integrity but increases network connection response time.

Prerequisites

The DB engine version and edition are one of the following:

- MySQL 8.0 Enterprise Edition
- MySQL 8.0 High-availability Edition
- MySQL 5.7 Enterprise Edition
- MySQL 5.7 High-availability Edition
- MySQL 5.6

Precautions

- The validity period of an SSL CA certificate is one year. You must renew the validity period of the SSL CA certificate in your application or client within one year. Otherwise, your application or client that uses an encrypted network connection cannot connect to RDS properly.
- SSL encryption increases CPU usage. Therefore, we recommend that you enable SSL encryption only for public endpoints when required. In typical cases, private endpoints do not require SSL encryption.
- Read/write splitting addresses do not support SSL encryption.
- Disabling SSL encryption restarts your RDS instance.

Enable SSL encryption

1. Log on to the RDS console.
2. In the upper-left corner, select the region where the target RDS instance is located.

3. Find the target RDS instance and click the instance ID.

4. In the left-side navigation pane, click Data Security.

5. Click the SSL Encryption tab.

6. Click the switch next to Disabled in the SSL Encryption parameter.

7. In the Configure SSL dialog box, select the endpoint for which you want to enable SSL encryption, then click OK.

Note:
You can choose to encrypt the private or public endpoint, but note that you can encrypt only one endpoint.

Select Protected Address:

- [ ] rm-1....mysql.rds.aliyuncs.com
- [ ] rm-2....mysql.rds.aliyuncs.com

Note: When the protected address is changed, the certificate automatically updates and your RDS instance is restarted.
8. Click **Download CA Certificate** to download the SSL CA certificate files in a compressed package.

The compressed package consists of the following three files:

- .p7b file: used to import CA certificate files in Windows operating systems.
- .pem file: used to import CA certificate files in other systems or applications.
- .jks file: used to import link CA certificate files in Java-based applications. The .jks file is stored in the TrustStore of Java.

**Note:**
When you use the .jks file in JDK 7 or JDK 8, you must modify the default JDK security configuration. Specifically, you must find the jre/lib/security/java.security file on the server where the database you want to access through SSL is located, and then reconfigure the file as follows:

```
jdk.tls.disabledAlgorithms=SSLv3, RC4, DH keySize < 224
```
jdbc.certpath.disabledAlgorithms=MD2, RSA keySize < 1024

If you do not modify the JDK security configuration, the system reports errors similar to the following:

javax.net.ssl.SSLHandshakeException: DHPublicKey does not comply to algorithm constraints

Configure the SSL CA certificate

After SSL encryption is enabled, you must configure the SSL CA certificate for your application or client when connecting to RDS. This section uses MySQL Workbench as an example to describe how to install the SSL CA certificate.

1. Start MySQL Workbench.
2. Choose **Database > Manage Connections**.
3. Enable **Use SSL** and import the SSL CA certificate files.

Renew the validity period of the SSL CA certificate

**Note:**

This operation causes your RDS instance to restart. You must make proper service arrangements before this operation.
Disable SSL encryption

1. Log on to the RDS console.
2. In the upper-left corner, select the region where the target RDS instance is located.
3. Find the target RDS instance and click the instance ID.
4. In the left-side navigation pane, click Data Security.
5. Click the SSL Encryption tab.
6. Click the switch next to Enabled in the SSL Encryption parameter. In the displayed Disable SSL Encryption dialog box, click OK.

22.4 Configure TDE for an ApsaraDB RDS for MySQL instance

Transparent Data Encryption (TDE) encrypts and decrypts data in real time when files are written or read. It encrypts data when files are written to disks and decrypts data when files are loaded into memory from disks. TDE does not increase the sizes of data files. You can use TDE without the need to change applications.

Prerequisites

- Your RDS instance runs one of the following MySQL versions and RDS editions:
  - MySQL 8.0 High-availability Edition (with local SSDs)
  - MySQL 5.7 High-availability Edition (with local SSDs)
  - MySQL 5.6
- Key Management Service (KMS) is activated. If KMS is not activated, you can activate it when you enable TDE.

Context

Encryption keys are created and managed by KMS. RDS does not provide the keys and certificates that are required for encryption. For specific zones, you can use the keys that are automatically generated by Alibaba Cloud or use your own key materials to generate data keys, and then authorize your RDS instance to use these keys.

Note:

After TDE is enabled, the encryption algorithm AES_128_ECB is used.
Precautions

- Enabling TDE restarts your RDS instance and terminates all of its connections. Make appropriate service arrangements before you enable TDE. Proceed with caution.
- After TDE is enabled, it cannot be disabled.
- After TDE is enabled, you cannot change the key.
- After TDE is enabled, if you want to restore data to your computer, you must decrypt data on your RDS instance.
- After TDE is enabled, CPU utilization significantly increases.
- If you use an existing custom key, note the following points:
  - If you disable a key, set a key deletion plan, or delete the key materials, the key becomes unavailable.
  - After you revoke the key that is authorized for an RDS instance, the RDS instance becomes unavailable after it is restarted.
  - You must use an Alibaba Cloud account or an account that has the AliyunSTSA assumeRoleAccess permission.

Note:
For more information, see What is KMS?

Use a key that is automatically generated by Alibaba Cloud

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target instance and click its ID.
4. In the left-side navigation pane, click Data Security.
5. Click the **TDE** tab. Then, turn on **TDE Status**.

![TDE Settings](image)

After TDE is enabled, you must execute the following DDL statements on the MySQL table to encrypt or decrypt the data:
- Encrypt: `ALTER TABLE table_name ENCRYPTION='Y';`
- Decrypt: `ALTER TABLE table_name ENCRYPTION='N';`

6. In the dialog box that appears, select **Use an Automatically Generated Key** and click **OK**.

![Note](image)

**Note:**
After TDE is enabled, the encryption algorithm AES_128_ECB is used.

**Use an existing custom key**

1. Log on to the **ApsaraDB for RDS console**.

2. In the top navigation bar, select the region where the target RDS instance resides.

3. Find the target instance and click its ID.

4. In the left-side navigation pane, click **Data Security**.
5. Click the **TDE** tab. Then, turn on **TDE Status**.

6. In the dialog box that appears, select **Use an Existing Custom Key** and click **OK**.

   **Note:**
   If you do not have a custom key, click **create a key** to go to the KMS console and import the key materials. For more information, see KMS.

   ![Database TDE Settings](image)

**Encrypt a table**

Log on to the target database and execute one of the following statements to encrypt a table:
ApsaraDB for RDS

RDS MySQL Database / 22 Data security

• MySQL 5.6

   alter table <tablename> engine=innodb,block_format=encrypted;

• MySQL 5.7 or MySQL 8.0

   alter table <tablename> encryption='Y';

Decrypt a table

Execute one of the following statements to decrypt a table that is encrypted with TDE:

• MySQL 5.6

   alter table <tablename> engine=innodb,block_format=default;

• MySQL 5.7 or MySQL 8.0

   alter table <tablename> encryption='N';

FAQ

• Q: Can common database tools such as Navicat be used after TDE is enabled?

   A: Yes, common database tools such as Navicat can be used after TDE is enabled.

• Q: Why is data still in plaintext after it is encrypted?

   A: Data is stored in ciphertext. However, when you query it, the data is decrypted and then loaded into memory in plaintext. After TDE is enabled, data is not leaked even if backup files are disclosed. The backup files are encrypted and cannot be used to restore data to your computer. If you want to restore data to your computer, you must first decrypt data.

References

Configure TDE for an ApsaraDB RDS for SQL Server instance

Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_344</td>
<td>Enables TDE for an RDS instance.</td>
</tr>
</tbody>
</table>

22.5 Configure disk encryption for an ApsaraDB RDS for MySQL instance

This topic describes how to configure disk encryption for an ApsaraDB RDS for MySQL instance equipped with standard or enhanced SSDs. The disk encryption feature encrypts
the data on each disk of your RDS instance by using block storage. This way, your data cannot be cracked even if it is leaked.

For more information about disk encryption in other database engines, see the following topics:

- Configure disk encryption for an ApsaraDB RDS for SQL Server instance
- Configure disk encryption for an ApsaraDB RDS for PostgreSQL instance

Prerequisites

- Your RDS instance is being created. Disk encryption cannot be enabled after your RDS instance is created. For more information, see Create an ApsaraDB RDS for MySQL instance.
- The Standard SSD or Enhanced SSD storage type is selected for your RDS instance. For more information, see #unique_348.
- The High-availability Edition is selected for your RDS instance. For more information, see ApsaraDB for RDS edition overview.
- Your RDS instance resides in one of the following regions:
  - China (Hangzhou)
  - China (Shanghai)
  - China (Qingdao)
  - China (Beijing)
  - China (Shenzhen)
  - China (Hong Kong)
  - Singapore
  - Malaysia (Kuala Lumpur)
  - Indonesia (Jakarta)
  - Germany (Frankfurt)

Billing

The disk encryption feature is free of charge. You do not need to pay additional fees for the read or write operations you perform on encrypted disks.

Precautions

- Disk encryption cannot be disabled after you enable it.
• After you enable disk encryption for your RDS instance, both the snapshots generated by that RDS instance and the new RDS instances created from those snapshots are automatically encrypted.

• Disk encryption does not interrupt your business and you do not need to modify your application.

Procedure

Create an ApsaraDB RDS for MySQL instance with the Standard SSD or Enhanced SSD storage type and the Disk Encryption option selected. Then, select a key used for data encryption.

Note:
For more information, see Manage CMKs.

22.6 Data security best practices in financial scenarios

RDS MySQL provides comprehensive protection to guarantee data security in financial usage scenarios. There are multiple methods such as disaster recovery and backup, network isolation, and permission control that can guarantee the security of databases.

Disaster recovery and backup

Enterprise Edition instances

To further meet the high reliability and data security requirements in business scenarios, ApsaraDB for RDS provides the Enterprise Edition. In this edition, each instance serves as a master instance and has two slave instances. Multi-copy replication guarantees strong data consistency and provides financial-level reliability.

You can select the Enterprise Edition when you create an RDS instance. For more information, see Create an ApsaraDB RDS for MySQL instance.

Multiple zones

Each region where an ApsaraDB for RDS instance resides contains multiple zones. The zones in the same region have an extremely low network latency (less than 3 ms), and a fault in one zone does not impact the services in other zones. Multi-zone instances are
deployed on physical servers in different zones. When a zone fails, traffic can be quickly switched to another zone. The entire switchover process is transparent, and does not require any changes to be made to the application code.

You can set the zone to Multi-zone when you create an instance. For more information, see Create an ApsaraDB RDS for MySQL instance.

Permission control

RAM user authorization

Resource Access Management (RAM) allows you to create and manage RAM user accounts and control their operation permissions on resources of your Alibaba Cloud account. When multiple users in your enterprise need to manage resources at the same time, you can use RAM to assign the least privileges to users and avoid sharing your account key with other users. This reduces information security risks of your enterprise.

For more information, see #unique_351.

Database account authorization

ApsaraDB for RDS can authorize database accounts to manage databases based on the business needs of the production environment.

You can use the console to create an account and authorize the account database management permissions. For more information, see Create databases and accounts for an ApsaraDB RDS for MySQL instance.

Network isolation

VPC

ApsaraDB for RDS supports multiple network types. We recommend that you use VPC.

VPC is an isolated network environment with higher security and performance than classic networks. You must create a VPC in advance. For more information, see Create a default VPC and VSwitch.

If an RDS instance is deployed in a classic network, you can switch the network type of the instance to VPC. For more information, see #unique_107. If your RDS instance is deployed in a VPC, no configuration is required.

IP address whitelist
After an RDS instance is created, the default IP address whitelist is 127.0.0.1, which indicates that no device can access the RDS instance. You must manually add an IP address before you can connect to the RDS instance.

For more information, see Configure a whitelist for an ApsaraDB RDS for MySQL instance.

**Log audit**

**SQL explorer**

ApsaraDB for RDS provides the SQL explorer feature, so that you can perform security audit and performance diagnostics on your database.

For more information, see #unique_352.

**Data encryption**

**SSL encryption**

When you connect to a database through the public network, you can enable Secure Sockets Layer (SSL) encryption and install SSL CA certificates on the necessary applications and services. SSL is used on the transport layer to encrypt network connections. It increases the security and integrity of communication data, but it also increases the response time for network connection.

For more information, see #unique_138.

**Transparent Data Encryption**

Transparent Data Encryption (TDE) implements real-time I/O encryption and decryption for data files. It encrypts data before data is written to a disk and decrypts data before data is read from the disk. TDE does not increase the size of data files. You can use TDE without making changes to applications.

For more information, see #unique_137.
23 Audit

23.1 SQL Explorer

This topic describes how to use the SQL Explorer function of ApsaraDB RDS MySQL. Compared with the previous SQL audit function, SQL Explorer provides more diverse value-added services such as security audit and performance diagnosis at lower costs. Upgrading SQL audit to SQL Explorer does not affect your ApsaraDB RDS MySQL instance.

Prerequisites

Your RDS instance runs one of the following database engine versions and RDS editions:

- MySQL 8.0 High-availability Edition
- MySQL 5.7 High-availability Edition
- MySQL 5.6
- MySQL 5.5

Context

SQL Explorer records all data manipulation language (DML) and data definition language (DDL) operations through network protocol analysis and only consumes a small amount of CPU resources. The Trial Edition of SQL Explorer retains SQL log data generated within up to one day free of charge. If you want to retain your SQL log data for more than one day, you must pay for the storage resources you use.

Scenarios

- Your database is used for the finance, security, stocks, governmental affairs, or insurance sectors that have high requirements on data security.
- You need to analyze the running of your database to locate problems or verify the execution performance of SQL statements.
- Your data needs to be restored by using the SQL statements recorded in SQL log data.

Differences between SQL Explorer-generated logs and binary logs

Both SQL Explorer-generated logs and binary logs offer incremental data to your ApsaraDB RDS MySQL instance. However, the two types of logs differ in the following aspects:

- SQL Explorer-generated logs are similar to audit logs in MySQL and record all DML and DDL operations through network protocol analysis. SQL Explorer does not parse actual
parameter values. If a large amount of SQL statements are executed to query data, some operation records may be lost. As a result, the incremental data obtained through SQL Explorer-generated logs may be inaccurate.

- Binary logs record all add, delete, and modify operations and the incremental data used for data restoration. After binary logs are generated, they are temporarily stored in your ApsaraDB for RDS instance. The system periodically transfers every binary log file whose size reaches the specified threshold to OSS. In OSS buckets, these log files can be retained for up to seven days. However, a binary log file cannot be transferred if data is being written to it. Therefore, you may find that some binary log files fail to be uploaded to OSS. Binary logs are not generated in real time, but you can obtain accurate incremental data from them.

**Precautions**

- The time range for an online query extends up to 24 hours. SQL Explorer records all database-related operations, which involve a large amount of SQL statements. If the time range selected exceeds 24 hours, the query takes a long time and may even time out.

**Note:**

If you want to query SQL log data from a time range spanning longer than 24 hours, we recommend that you export the SQL log data asynchronously to your computer.

- Online query supports combination query. For example, enter test1 test2 in the keyword search field to query the SQL log containing test1 or test2.
- Fuzzy query is not supported for online query.
- The length of SQL statement is limited to 2000 bytes, and the exceeding part cannot be recorded.
Function description

- SQL logging

  Records all database-related operations. The recorded information helps you locate database problems, analyze actions, and perform security audits.

- Advanced search

  Allows you to search for data in various dimensions such as database, thread ID, user, client IP address, execution time, or number of scanned rows. You can export and download the search results.

Note:

- If you search for data in a single dimension, you can specify more than one search criterion and the system considers the specified search criteria in OR relationships. For example, if you specify two search criteria, user1 and user2, in the Users field, the system displays all SQL statements executed by user1 and user2.

- If you search for data in more than one dimension, the system considers the specified dimensions in AND relationships. For example, if you enter user1 in the Users field and enter SELECT in the Operation Type field, the system displays all SELECT statements executed by user1.
- Fuzzy match is not supported.
• SQL analysis

Allows you to analyze SQL logs generated within the specified time range. Using the SQL log analysis results, you can identify SQL statements that cause database performance issues.

• Cost efficiency

The column-oriented database storage and compression technology used reduces SQL log space usage and saves 60% of overall storage costs.

**Enable SQL Explorer**

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID.

4. In the left-side navigation pane, click **SQL Explorer**.

5. Click **Activate Now**.

6. Specify the SQL log storage duration and click **Activate**.

**Note:**
If the storage duration of an SQL log file elapses, the system deletes the file.

- Trial Edition: SQL Explorer remains available for a long time. However, SQL log files are retained for only one day. You can query SQL log data during up to one day it
ApsaraDB for RDS

is retained. You cannot use advanced functions such as data export, nor is data integrity guaranteed.

- **Paid Edition**: SQL log files can be retained for 30 days, 6 months, 1 year, 3 years, or 5 years. You are charged an hourly fee of USD 0.0018/GB.

### Change SQL log storage duration

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the page, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click **SQL Explorer**.
5. Click **Service Settings**.
6. On the Service Settings page that appears, change the SQL log storage duration and click OK.

![Service Settings](image)

### Disable SQL Explorer

**Note:**

After you disable SQL Explorer, all existing SQL log files are deleted. We recommend that you export and save SQL log files to your computer before disabling SQL Explorer.

1. Log on to the ApsaraDB for RDS console.

2. In the upper-left corner of the page, select the region where the target RDS instance resides.

3. Find the target RDS instance and click its ID.

4. In the left-side navigation pane, click **SQL Explorer**.
5. Click **Export**.

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

7. After the export is complete, click **View Exported List** and download SQL log files to your computer.

8. Click **Service Settings**.
9. Turn off the Activate SQL Explorer switch.

### FAQ

**How do I view the SQL log size after enabling SQL Explorer?**

Log on to the ApsaraDB for RDS console. Find the target RDS instance and navigate to its **Basic Information** page. In the **Usage Statistics** section, view the SQL log size.

### 23.2 Manage logs

This topic describes how to manage error logs, slow query logs, and primary/secondary instance switching logs of an ApsaraDB for RDS MySQL instance through the ApsaraDB for RDS console. The logs help you locate faults.

**Note:**

All RDS editions support the log management function. For more information about binary logs, see [Back up an ApsaraDB RDS for MySQL instance](#) and [Download data backup files and binary log files](#).

For more information about how to manage logs for other storage engines, see the following topics:

- SQL Server log management
- Manage PostgreSQL logs
- PPAS log management
- Log Management for MariaDB TX
Procedure

1. Log on to the ApsaraDB for RDS console.
2. In the upper-left corner of the console, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID.
4. In the left-side navigation pane, click Logs.
5. On the Logs page that appears, click the Error Log, Slow Query Log, Slow Query Log Summary, or Primary/Secondary Instance Switching Log tab, select a time range, and click Search.

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Log</td>
<td>Records database running errors that occurred within the last month.</td>
</tr>
<tr>
<td>Slow Query Log</td>
<td>Provides SQL statements that each took more than 1 second to run within the last month. Duplicate SQL statements are deleted. You can change this 1-second threshold by reconfiguring the long_query_time parameter. For more information, see Reconfigure parameters for an RDS MySQL instance.</td>
</tr>
<tr>
<td>Slow Query Log Summary</td>
<td>Provides statistics and analysis reports on SQL statements that each took more than 1 second to run within the last month. You can change this 1-second threshold by reconfiguring the long_query_time parameter. For more information, see Reconfigure parameters for an RDS MySQL instance.</td>
</tr>
</tbody>
</table>

Note:
Slow query logs in the ApsaraDB for RDS console are updated once every minute. Optionally, you can query real-time slow query logs from the mysql.slow_log table.
<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary/Secondary Instance Switching Log</td>
<td>Records switchovers between the primary and secondary instances triggered within the last month. This tab appears only if your RDS MySQL instance is not in the Basic Edition.</td>
</tr>
</tbody>
</table>

**Note:**
If your ApsaraDB for RDS MySQL instance resides in the China (Zhangjiakou-Beijing Winter Olympics) region, the system retains only error logs, slow query logs, and slow query log summary generated within the last nine days.

### 23.3 View the slow query logs of an ApsaraDB RDS for MySQL instance

This topic describes how to view the slow query logs of an ApsaraDB RDS for MySQL instance.

**View slow query logs in the ApsaraDB for RDS console**

1. Log on to the ApsaraDB for RDS console.
2. In the top navigation bar, select the region where the target RDS instance resides.
3. Find the target RDS instance and click its ID. The Basic Information page appears.
4. In the left-side navigation pane, click Logs.
5. Click the Slow Query Logs tab and view details about the slow query logs of the RDS instance.

**View slow query logs by using commands**

**Note:**
If your RDS instance is running MySQL 5.5, you cannot view slow query logs by using commands.

1. Log on to the target RDS instance.
2. In the top navigation bar, choose SQL Operations > SQL Window.
3. Run the following command to obtain the slow query logs of the RDS instance:

   ```sql
   select * from mysql.slow_log
   ```

### 23.4 View the event history of an ApsaraDB RDS for MySQL instance

This topic describes how to view the operation and maintenance (O&M) events that are performed by users and Alibaba Cloud on an ApsaraDB RDS for MySQL instance. These events include instance creation and parameter reconfiguration.

**Billing**

The event history feature is free of charge in the public preview phase, but starts to be charged after the public preview phase ends.

**Scenarios**

- Track instance management operations.
- Audit the security of instance management operations.
- Audit the compliance of the instance management operations that are performed by Alibaba Cloud. This applies to security-demanding sectors, such as finance and government affairs.

**View the event history feature**

1. Log on to the ApsaraDB for RDS console.

2. In the top navigation bar, select the region for which you want to enable the event history feature.

3. In the left-side navigation pane, click **Event History**.
Introduction to the Event History page

The Event History page shows details about historical events in the selected region. These details include the resource type, resource name, and event type. The following table describes the parameters of a historical event.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Type</td>
<td>The type of the RDS resource managed in the event. Only the <strong>Instance</strong> resource type is supported.</td>
</tr>
<tr>
<td>Resource Name</td>
<td>The name of the RDS resource managed in the event. If the value of the <strong>Resource Type</strong> parameter is <strong>Instance</strong>, the <strong>Resource Name</strong> column displays the ID of the involved RDS instance.</td>
</tr>
<tr>
<td>Event Type</td>
<td>The type of the event, for example, <strong>Instance Management</strong>, <strong>Database Management</strong>, <strong>Read-write Splitting</strong>, and <strong>Network Management</strong>. For more information, see <strong>Events</strong>.</td>
</tr>
<tr>
<td>Event Name</td>
<td>The operation executed in the event. For example, if the event type is <strong>Instance Management</strong>, supported operations include <strong>Create Instance</strong>, <strong>Delete Instance</strong>, <strong>Change Specifications</strong>, and <strong>Restart Instance</strong>. For more information, see <strong>Events</strong>.</td>
</tr>
<tr>
<td>Run At</td>
<td>The time when the event was executed.</td>
</tr>
<tr>
<td>User Type</td>
<td>The initiator of the event. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• User: initiates operations by using the ApsaraDB for RDS console or the API.</td>
</tr>
<tr>
<td></td>
<td>• System: initiates automatic O&amp;M operations or periodic tasks.</td>
</tr>
<tr>
<td></td>
<td>• O&amp;M Administrator: initiates manual O&amp;M operations.</td>
</tr>
<tr>
<td>Cause</td>
<td>The cause of the event. Valid values:</td>
</tr>
<tr>
<td></td>
<td>• User Action: The event was initiated from a user by using the ApsaraDB for RDS console or the API.</td>
</tr>
<tr>
<td></td>
<td>• System Action or O&amp;M Action: The event was initiated from the system or an O&amp;M administrator.</td>
</tr>
<tr>
<td>Parameters</td>
<td>The request parameters used by a user to initiate the event in the ApsaraDB for RDS console.</td>
</tr>
</tbody>
</table>
The Event History page shows the historical events that are generated about 5 minutes earlier.

Historical events are presented specific to regions. You can select the target region in the top navigation bar and then view the historical events in the selected region.

<table>
<thead>
<tr>
<th>Event type</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Management</td>
<td>Restart Instance (RestartDBInstance)</td>
</tr>
<tr>
<td></td>
<td>Renew (RenewInstance)</td>
</tr>
<tr>
<td></td>
<td>Change Specifications (ModifyDBInstanceSpec)</td>
</tr>
<tr>
<td></td>
<td>Migrate Across Zones (MigrateToOtherZone)</td>
</tr>
<tr>
<td></td>
<td>Shrink Log (PurgeDBInstanceLog)</td>
</tr>
<tr>
<td></td>
<td>Upgrade Kernel Version (UpgradeDBInstanceEngineVersion)</td>
</tr>
<tr>
<td></td>
<td>Modify Instance Description (ModifyDBInstanceIdationDescription)</td>
</tr>
<tr>
<td></td>
<td>Modify Maintenance Window (ModifyDBInstanceMaintainTime)</td>
</tr>
<tr>
<td></td>
<td>Create Read-only Instance (CreateReadOnlyDBInstance)</td>
</tr>
<tr>
<td></td>
<td>Destroy Instance (DestroyDBInstance)</td>
</tr>
<tr>
<td></td>
<td>Modify Upgrade Mode of Kernel Version (ModifyDBInstanceAutoUpgradeMinorVersion)</td>
</tr>
<tr>
<td></td>
<td>Edit Parameters (ModifyParameter)</td>
</tr>
<tr>
<td>CloudDBA</td>
<td>Create Diagnostics Report (CreateDiagnosticReport)</td>
</tr>
<tr>
<td>Database Management</td>
<td>Create Database (CreateDatabase)</td>
</tr>
<tr>
<td></td>
<td>Delete Database (DeleteDatabase)</td>
</tr>
<tr>
<td></td>
<td>Modify Database Description (ModifyDBDescription)</td>
</tr>
<tr>
<td>Event type</td>
<td>Operation</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Replicate Database Between Instances (CopyDatabaseBetweenInstances)</td>
</tr>
<tr>
<td></td>
<td>Modify System Collation and Time Zone (ModifyCollationTimeZone)</td>
</tr>
<tr>
<td>Read-write Splitting</td>
<td>Create Read-write Splitting Endpoint (AllocateReadWriteSplittingConnection)</td>
</tr>
<tr>
<td></td>
<td>Query System-assigned Weight (CalculateDBInstanceWeight)</td>
</tr>
<tr>
<td></td>
<td>Modify Read-write Splitting Policy (ModifyReadWriteSplittingConnection)</td>
</tr>
<tr>
<td></td>
<td>Release Read-write Splitting Endpoint (ReleaseReadWriteSplittingConnection)</td>
</tr>
<tr>
<td>Security Management</td>
<td>Enable Enhanced Whitelist (MigrateSecurityIPMode)</td>
</tr>
<tr>
<td></td>
<td>Enable SSL (ModifyDBInstanceSSL)</td>
</tr>
<tr>
<td></td>
<td>Enable TDE (ModifyDBInstanceTDE)</td>
</tr>
<tr>
<td></td>
<td>Modify Whitelist (ModifySecurityIps)</td>
</tr>
<tr>
<td>Account Management</td>
<td>Create Account (CreateAccount)</td>
</tr>
<tr>
<td></td>
<td>Delete Account (DeleteAccount)</td>
</tr>
<tr>
<td></td>
<td>Authorize Account to Access Database (GrantAccountPrivilege)</td>
</tr>
<tr>
<td></td>
<td>Revoke Database Permissions from Account (RevokeAccountPrivilege)</td>
</tr>
<tr>
<td></td>
<td>Modify Description of Database Account (ModifyAccountDescription)</td>
</tr>
<tr>
<td></td>
<td>Reset Account Password (ResetAccountPassword)</td>
</tr>
<tr>
<td></td>
<td>Reset Permissions of Superuser Account (ResetAccount)</td>
</tr>
<tr>
<td>High Availability (HA)</td>
<td>Trigger Switchover Between Primary and Secondary Instances (SwitchDBI instanceHA)</td>
</tr>
<tr>
<td></td>
<td>Modify HA Mode (ModifyDBInstanceHAConfig)</td>
</tr>
<tr>
<td>Network Management</td>
<td>Apply for Public Endpoint (AllocateInstancePublicConnection)</td>
</tr>
<tr>
<td></td>
<td>Modify Expiry Time of Endpoint (ModifyDBInstanceNetworkExpireTime)</td>
</tr>
<tr>
<td>Event type</td>
<td>Operation</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Modify Endpoint and Port (ModifyDBInstanceConnectionString)</td>
</tr>
<tr>
<td></td>
<td>Switch Network Type (ModifyDBInstanceNetworkType)</td>
</tr>
<tr>
<td></td>
<td>Release Public Endpoint (ReleaseInstancePublicConnection)</td>
</tr>
<tr>
<td></td>
<td>Switch Between Internal and Public Endpoints (SwitchDBInstanceNetType)</td>
</tr>
<tr>
<td></td>
<td>Enable/disable Log Audit (ModifySQLCollectorPolicy)</td>
</tr>
<tr>
<td>Backup Restoration</td>
<td>Create Data Backup (CreateBackup)</td>
</tr>
<tr>
<td></td>
<td>Clone Instance (CloneDBInstance)</td>
</tr>
<tr>
<td></td>
<td>Create Temporary Instance (CreateTempDBInstance)</td>
</tr>
<tr>
<td></td>
<td>Modify Backup Policy (ModifyBackupPolicy)</td>
</tr>
<tr>
<td></td>
<td>Restore Backup Set to Original Instance (RestoreDBInstance)</td>
</tr>
<tr>
<td></td>
<td>Delete Data Backup (DeleteBackup)</td>
</tr>
<tr>
<td></td>
<td>Restore Database (RecoveryDBInstance)</td>
</tr>
<tr>
<td>Cross-region Backup</td>
<td>Restore Data to New Instance Across Regions (CreateDdrInstance)</td>
</tr>
<tr>
<td>Restoration</td>
<td>Modify Cross-region Backup Settings (ModifyInstanceCrossBackupPolicy)</td>
</tr>
<tr>
<td>SQL Server Backup Migration</td>
<td>Restore Backup File in OSS to RDS Instance (CreateMigrateTask)</td>
</tr>
<tr>
<td>to Cloud</td>
<td>Make Database Available While Migrating Backup Data to Cloud (CreateOnlineDatabaseTask)</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Set Monitoring Frequency (ModifyDBInstanceMonitor)</td>
</tr>
<tr>
<td>Data Migration</td>
<td>Create Upload Path for SQL Server (CreateUploadPathForSQLServer)</td>
</tr>
<tr>
<td></td>
<td>Import Data from Other RDS (ImportDatabaseBetweenInstances)</td>
</tr>
<tr>
<td></td>
<td>Cancel Migration Task (CancelImport)</td>
</tr>
<tr>
<td>Tag Management</td>
<td>Bind Tags to Instance (AddTagsToResource)</td>
</tr>
<tr>
<td></td>
<td>Remove Tag (RemoveTagsFromResource)</td>
</tr>
</tbody>
</table>
Related operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_360</td>
<td>Queries the events of an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_361</td>
<td>Queries the status of the event history feature of an ApsaraDB for RDS instance.</td>
</tr>
<tr>
<td>#unique_362</td>
<td>Enables or disables the event history feature of an ApsaraDB for RDS instance.</td>
</tr>
</tbody>
</table>

23.5 Configure local binary log policies for an RDS MySQL instance

This topic describes how to configure local binary log policies for an RDS MySQL instance. RDS MySQL deletes local binary log files according to the polices you specify. Before you set the policies, you must learn about the rules for MySQL to generate and delete binary log files.

Context

For information about the related API, see #unique_364.

The rules for an RDS MySQL instance to generate binary log files are as follows:

- In typical cases, when the size of the current binary log file exceeds 500 MB, the system starts to write data into a new binary log file. When the system is writing data into the new file, the old file is uploaded asynchronously.
- In certain circumstances, the system stops writing data into the current binary log file due to reasons such as command execution and system restart even when the file size does not exceed 500 MB.
- In certain circumstances, the size of a binary log file exceeds 500 MB when, for example, the system is executing a large transaction.

The default rules for an RDS MySQL instance to delete binary log files are as follows:

- By default, the system retains the binary log files generated within the last 18 hours.
- When the space usage does not exceed 80%, the system retains the binary log files that occupy 30% of the space regardless of whether these files have been uploaded to OSS or not.
When the space usage exceeds 80%, the system initiates requests for deleting local binary log files after these files are uploaded to OSS. The requests must be scheduled, and therefore these files are deleted at a certain delay.

Procedure

1. Log on to the RDS console.

2. In the upper-left corner, select the region where the target RDS instance is located.

3. In the left-side navigation pane, click Backup and Restoration.

4. Click the Local Log Settings tab and in the Local Binary Log Settings section view the local log settings of the RDS instance.
5. Click Edit.

![Local Binary Log Settings dialog box](image)

- **Retention Period**: The period for retaining each generated binary log file. After the retention period elapses, the system deletes the binary log file after backing up it. Note that the system backs up binary log files only after you enable the log backup function. Value range: 0 - 7*24. Default value: 18. Unit: hour.

- **Max Space Usage**: The maximum space usage that is allowed for local binary log files. When the space occupied by local binary log files exceeds the maximum usage, the system starts to delete earlier local binary log files. The system stops deleting local binary log files until the space usage drops below the maximum usage. Value range: 0 - 50%. Default value: 30%.

- **Protection Available Space**: Indicates whether the system deletes earlier local binary log files when the total space usage of the RDS instance exceeds 80% or the

6. In the Local Binary Log Settings dialog box, set Retention Period, Max Space Usage, and Protection Available Space

Parameter description:

- **Retention Period**: The period for retaining each generated binary log file. After the retention period elapses, the system deletes the binary log file after backing up it. Note that the system backs up binary log files only after you enable the log backup function. Value range: 0 - 7*24. Default value: 18. Unit: hour.

- **Max Space Usage**: The maximum space usage that is allowed for local binary log files. When the space occupied by local binary log files exceeds the maximum usage, the system starts to delete earlier local binary log files. The system stops deleting local binary log files until the space usage drops below the maximum usage. Value range: 0 - 50%. Default value: 30%.

- **Protection Available Space**: Indicates whether the system deletes earlier local binary log files when the total space usage of the RDS instance exceeds 80% or the
remaining space is less than 5 GB. The system stops deleting local binary log files when the total space usage drops below 80% and the remaining space is greater than 5 GB. Default value: Enable.

7. After you set the parameters, click **OK**.
24 Tag

24.1 Create tags

This topic describes how to create tags for one or more RDS instances. If you have a large number of RDS instances, you can create tags and then bind the tags to the instances so that you can classify and better manage the instances. Each tag consists of a key and a value.

Limits

- Up to 10 tags can be bound to each RDS instance, and each tag must have a unique key. Tags with the same key are overwritten.
- You can bind up to five tags at a time.
- Tag information is independent in different regions.
- After you unbind a tag from an RDS instance, the tag is deleted if it is not bound to any other RDS instance.

Procedure

1. Log on to the RDS console and in the left-side navigation pane, click **Instances**.
2. In the upper-left corner, select the region where the target RDS instance is located.
3. Specify the method of adding tags.

- If you want to add tags to only one RDS instance, find the RDS instance and in the **Actions** column choose **More > Edit Tag**.
- If you want to add tags to more than one RDS instance, select the RDS instances and click **Edit Tag**.

4. Click **Add**, enter the **Key** and **Value**, and click **Confirm**.

**Note:**
If you have already created tags, you can click **Available Tags** and select an existing tag.

5. After you add all the tags you need, click **Confirm**.
ApsaraDB for RDS

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddTagsToResource</td>
<td>Used to bind a tag to RDS instances.</td>
</tr>
</tbody>
</table>

### 24.2 Delete tags

This topic describes how to delete tags from an RDS instance when you no longer need the tags or due to adjustments to the instance.

#### Limits

After you unbind a tag from an RDS instance, the tag is deleted if it is not bound to any other instance.

#### Procedure

1. Log on to the [RDS console](https://console.aliyun.com/) and in the left-side navigation pane, click **Instances**.
2. In the upper-left corner, select the region where the target RDS instance is located.
3. Find the target RDS instance and in the **Actions** column, choose **More > Edit Tag**.
4. Find the tag you want to delete, and click the X button following the tag.

5. Click Confirm.

### APIs

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#unique_367</td>
<td>Used to unbind a tag from an RDS instance.</td>
</tr>
</tbody>
</table>

#### 24.3 Filter RDS instances by tag

This topic describes how to filter RDS instances by tag.

1. Log on to the RDS console.

2. In the upper-left corner, select the region where the target RDS instance is located.
3. On the **Basic Information** tab, click the **Tag** button next to **Search** and select a tag key and a tag value.

    **Note:**
    You can click the X button following the tag key to cancel the filter operation.

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DescribeTags</td>
<td>Used to query tags.</td>
</tr>
</tbody>
</table>
25 Appendixes

25.1 Commonly used SQL statements for MySQL

This topic lists some of the commonly used SQL statements.

For more information about the SQL statements including parameters and restrictions, see MySQL 5.7 Reference Manual.

Database-related SQL statements

<table>
<thead>
<tr>
<th>Operation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a database and designate a character set.</td>
<td>create database db01 DEFAULT CHARACTER SET gbk COLLATE gbk_chinese_ci;</td>
</tr>
<tr>
<td>Delete a database.</td>
<td>drop database db01;</td>
</tr>
</tbody>
</table>

Account-related SQL statements

Note:
If an RDS instance has a premier account, the passwords of the other accounts under this instance cannot be changed by using the premier account. To change the password of another account, you must delete this account and create a new one.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an account.</td>
<td><code>CREATE USER 'username'@'host' IDENTIFIED BY 'password';</code></td>
</tr>
<tr>
<td>Delete an account.</td>
<td><code>DROP USER 'username'@'host';</code></td>
</tr>
<tr>
<td>Authorize the account.</td>
<td><code>GRANT SELECT ON db01. * TO 'username'@'host';</code></td>
</tr>
<tr>
<td>Query the created accounts in the database.</td>
<td><code>SELECT user,host,password FROM mysql.user_view;</code> or <code>show grants for xxx</code></td>
</tr>
<tr>
<td>Reclaim permissions.</td>
<td>• Reclaim all permissions:</td>
</tr>
<tr>
<td></td>
<td><code>REVOKE ALL PRIVILEGES,GRANT OPTION FROM 'username'@'host';</code></td>
</tr>
<tr>
<td></td>
<td>• Reclaim specific permissions:</td>
</tr>
<tr>
<td></td>
<td><code>REVOKE UPDATE ON *. * FROM 'username'@'host';</code></td>
</tr>
</tbody>
</table>

25.2 Grant backup file download permissions to a RAM user with only read-only permissions

This topic describes how to grant backup file download permissions to a RAM user who only has read-only permissions. For security purposes, a typical RAM user cannot download backup files.

Procedure

1. Log on to the RAM console.
2. In the left-side navigation pane, choose Permissions > Policies.
3. Click **Create Policy** and set the parameters.

The policy content is as follows:

```json
{
   "Statement": [
      {
         "Effect": "Allow",
         "Action": ["rds:Describe*", "rds:ModifyBackupPolicy"],
         "Resource": "*"
      }
   ],
   "Version": "1"
}
```

4. Click **OK**.

5. In the left-side navigation pane, choose **Permissions > Grants**.

6. Click **Grant Permission** and add the new permission policy to the target RAM user.
25.3 Authorize RDS to access KMS

RDS must be able to access Key Management Service (KMS) before SSD encryption can be enabled. This topic describes how to authorize access to KMS in the RAM console.

Authorization procedure

1. Visit the Policies page in the RAM console.
2. Click Create Policy, and follow these steps to create a permission policy:

   ![Note]

   A permission policy is a set of permissions that are described in syntax structures.
   A policy specifies the authorized resources, authorized actions, and authorization conditions.

   a) Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy Name</td>
<td>Enter the name of the permission policy to be created. The name of the permission policy must be unique.</td>
</tr>
<tr>
<td>Note</td>
<td>Enter information to better identify the permission policy.</td>
</tr>
<tr>
<td>Configuration Mode</td>
<td>Select a configuration mode.</td>
</tr>
<tr>
<td></td>
<td>• Visualized: Click Add Statement to specify the permission effect, product or service, and actions among others.</td>
</tr>
<tr>
<td></td>
<td>• Script: Select a policy schema to set the permission policy. You can copy the script from the following &quot;Note.&quot;</td>
</tr>
</tbody>
</table>

   ![Note]

   An example script is as follows:

   ```json
   {
     "Version": "1",
     "Statement": [
       {
         "Action": [
           "kms:List*",
           "kms:DescribeKey",
           "kms:TagResource"
         ]
       }
     ]
   }
   ```
b) Click OK.

3. In the left-side navigation pane, click RAM Roles.
4. Click **Create RAM Role**, and follow these steps to create a RAM role.

a) Select **Alibaba Cloud Service** for **Trusted entity type** and click **Next**.

b) Configure the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM Role Name</td>
<td>Enter the name of the RAM role to be created. The name of the RAM role must be unique.</td>
</tr>
<tr>
<td>Note</td>
<td>Enter information to better identify the RAM role.</td>
</tr>
<tr>
<td>Select Trusted Service</td>
<td>Select RDS from the drop-down list.</td>
</tr>
</tbody>
</table>

c) Click **OK**.

5. Click **Add Permissions to RAM role** under the message **The Role has been created**, and associate the permission policy that you created with the RAM role.

a) In the Select Policy section, select **Custom Policy** from the drop-down list, and in the search bar enter the name of the permission policy that you created. After you find the permission policy, click to add it to the **Selected** list.

b) Click **OK**.
What to do next

An Alibaba Cloud Resource Name (ARN) is the globally unique resource identifier of a RAM role. ARNs follow the naming conventions stipulated by Alibaba Cloud. For example, the ARN of the role named devops under an Alibaba Cloud account is `acs:ram::123456789012****:role/samplerole`.

1. Visit the RAM Roles page in the RAM console.
2. Find the target RAM role and click the role name.
3. In the upper-right corner of the page that appears, view the ARN of the RAM role.

25.4 Cached data persistence

ApsaraDB RDS can be used together with ApsaraDB Memcache and Redis to form storage solutions with high throughput and low delay. This document describes the cached data persistence solution based on the combined use of RDS and Memcache.

Background information

Compared with RDS, Memcache and Redis have the following features:

- Quick response: The request delay of ApsaraDB Memcache and Redis is usually within several milliseconds.
- The cache area supports a higher Queries Per Second (QPS) than RDS.
System requirements

- bmемcached (with support for SASL extension) has been installed in the local environment or ECS.

  bmемcached download address: Click Here to download.

  The bmемcached installation command is as follows:

  ```bash
  pip install python-binary-memcached
  ```

- Python is used as an example. Python and pip must be installed in the local environment or ECS.

Sample code

The following sample code realizes the combined use of ApsaraDB RDS and Memcache:

```python
/usr/bin/env python
import bmемcached
Memcache_client = bmемcached.Client(('ip:port'), 'user', 'passwd')
# Search for a value in ApsaraDB Memcache
res = os.client.get('test')
if res is not None:
    return res # Return the value found
else:
    # Query RDS if the value is not found
    res = mysql_client.fetchone(sql)
    Memcache_client.put('test', res) # Write cached data to ApsaraDB for Memcache
    return res
```

25.5 Multi-structure data storage

OSS is a cloud storage service provided by Alibaba Cloud, featuring massive capacity, security, low cost, and high reliability. RDS can work with OSS to form multiple types of data storage solutions.

For example, when the business application is a forum and RDS works with OSS, resources such as registered users' images and post content images can be stored in OSS to reduce the storage pressure of RDS.

Sample code

OSS works with the RDS.

1. Initialize OssAPI.

```python
from oss.oss_api import *
endpoint="oss-cn-hangzhou.aliyuncs.com"
accessKeyId, accessKeySecret="your id","your secret"
```
2. Create a bucket.

```python
# Set the bucket to private-read-write
res = oss.create_bucket(bucket, "private")
print "%s\n%s" % (res.status, res.read())
```

3. Upload an object.

```python
res = oss.put_object_from_file(bucket, object, "test.txt")
print "%s\n%s" % (res.status, res.getheaders())
```

4. Obtain the corresponding object.

```python
res = oss.get_object_to_file(bucket, object, "/filepath/test.txt")
print "%s\n%s" % (res.status, res.getheaders())
```

In the ECS application code, RDS stores the ID of each user, and OSS stores the avatar resource of the user. The Python code is as follows:

```python
#!/usr/bin/env python
from oss.oss_api import *
endpoint="oss-cn-hangzhou.aliyuncs.com"
accessKeyId, accessKeySecret="your id","your secret"
oss = OssAPI(endpoint, accessKeyId, accessKeySecret)
user_id = mysql_client.fetch_one(sql)# Search for user_id in RDS
# Obtain and download the user avatar to the corresponding path
oss.get_object_to_file(bucket, object, your_path/user_id+'.png')
# Process the uploaded user avatar
oss.put_object_from_file(bucket, object, your_path/user_id+.png')
```